Using the Random Nearest Neighbor data mining method to extract maximum information content from weather forecasts from multiple predictors of weather and one predictand (low-level turbulence).

Links to sections:

EXECUTIVE SUMMARY

A new methodology of data mining is developed to find relationships between Air Force Weather Agency (AFWA) WRF 15-km atmospheric model forecast data and low-level turbulence. Archives of historical model data forecast predictors at model gridpoints and verifying pilot reports (PIREPS) of turbulence have been collected. The new data mining method, Random Nearest Neighbor (RNN), will be shown to be capable of extracting nearly the maximum possible amount of information from a multiple predictor, single predictand dataset. Relationships between WRF model predictors and PIREPS were developed using the new data mining methodology.

The new methodology was inspired by the Random Forest (RF) method (Breiman 2001). The RF method recognizes weak points in the use of decision trees in forecasting. RF uses random numbers to modify or "perturb" decision trees. It creates approximately 500 decision trees, thus an "ensemble" of them, or a "forest" of trees, and uses either the average or the most frequent

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **30 OCT 2014**	2. REPORT TYPE **Unknown**	3. DATES COVERED **01 Jan 2014 - 30 Apr 2014**

4. TITLE AND SUBTITLE

Using the Random Nearest Neighbor data mining method to extract maximum information content from weather forecasts from multiple predictors of weather and one predictand (low-level turbulence).

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

David L. Keller

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Weather Agency

8. PERFORMING ORGANIZATION REPORT NUMBER

2014-038

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Using the Random Nearest Neighbor data mining method to extract maximum information content from weather forecasts from multiple predictors of weather and one predictand (low-level turbulence). (3 paragraphs omitted) Using the Random Nearest Neighbor data mining method to extract maximum information content from weather forecasts from multiple predictors of weather and one predictand (low-level turbulence).

15. SUBJECT TERMS

Random Nearest Neighbor, data mining, weather model turbulence, information extraction, forecasting, meteorology, pilot reports

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**50**	

result as the forecast value. The process is analogous to ensemble modeling known to Air Force and civilian numerical weather prediction.

The RNN approach of this report utilizes random numbers, but bypasses the creation of decision trees. In RNN, a random gridpoint is selected from the historical archive. “Similar” gridpoints in the archive are selected. This process is repeated until the historical archive is completely sampled. Each grouping of similar gridpoints is a “neighborhood”. The average amount of turbulence that occurs with each neighborhood is found from the historical archive. To make an operational forecast, the corresponding neighborhoods are applied to real-time model data. Thus RNN is a nearly direct method of looking up the historical amount of turbulence that should be forecast, without curve fitting or fitting data into an arbitrary data structure, while taking into account combinations of multiple predictors.

In this report, the RNN methodology is used to achieve nearly the best possible turbulence forecast from a domain consisting of predictors at model gridpoints and corresponding verification from PIREPS. Two experiments using RNN will demonstrate that RNN almost completely accomplishes the goal of accurately re-creating non-linear relationships of combinations of predictors with varying combinations of values. In the first experiment with real data, it will be seen that RNN accurately linearizes a predictor to the predictand. The second experiment uses a synthetic dataset. It will be seen that RNN accurately re-creates that synthetic dataset. RNN is then utilized with the real dataset. After demonstrating the effectiveness of the RNN methodology, it will be seen that low-level turbulence has limited forecastability using the turbulence dataset used in this study.

The goals of this technical report are three-fold: 1) to introduce RNN as a data mining methodology; 2) to demonstrate its effectiveness in extracting potentially complex non-linear multiple-predictor vs. predictand relationships, and 3) the implications of forecasting turbulence. Other facets of data mining and statistical forecasting, such as predictor selection techniques, are acknowledged but not explored in this report. An effort is made to explain clearly, to non-experts in statistics, how RNN works. Based on the real data results, reasons for limited forecastability of turbulence will be proffered.

1. Introduction

Aviation turbulence may the most difficult weather parameter to forecast (in the subjective opinion of the author). As subjective evidence, one can look at typical examples of low-level turbulence reports from PIREPS, such as Figures 1 and 6, and find little in the way of patterns. Neither Figure 1 nor Figure 6 suggest any obvious large-scale trough/ridge pattern, or frontal systems. An unpublished experiment by the author confirmed that turbulent PIREPS from *upper* levels are not randomly spaced, but do indeed have some geographical grouping beyond random chance. However, the statistical validation of the non-randomness of turbulence sheds little light on the meteorological causes. In an effort to understand the forecastability of turbulence, a methodology was sought to extract the most information possible from existing weather model forecast data.

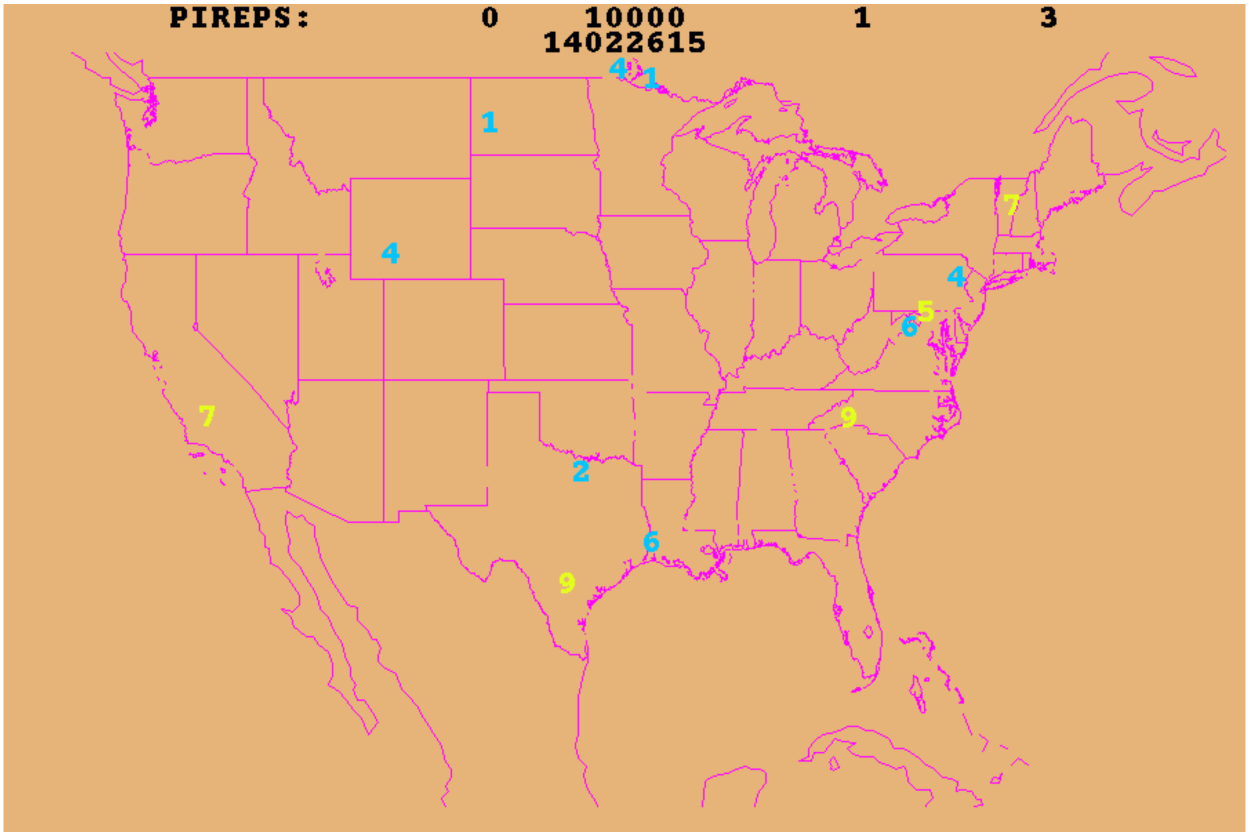


Figure 1. Elevation (in thousands of feet) of turbulence from PIREPS, within 1.5 hours of 2014 February 26, 15UTC, elevation 0 to 10,000 feet. Blue: light turbulence, yellow: moderate.

Several methodologies exist to create automated statistical or data-mined forecasts based on relationships between historical archives of model data and observations that are the target forecast. Regression or curve-fitting are one general type of such a forecast. Curve-fitting methodologies assume that weather events (temperature, lightning, precipitation, turbulence, icing, etc.) can be described with some method of curve-fitting. Data mining methods often utilize a data structure as a means of organizing data. Table 1 lists examples of each statistical methodology.

Table 1. List of common methods of regression and data mining methodologies used in physical sciences.

Regression	Data mining
Multiple linear regression	Contingency tables
Neural network	Decision trees
Logistic regression	Clustering
Principle Component Analysis	Self Organizing Maps

The goal of these statistical and data mining methodologies is to consider many predictors, ranging from several to perhaps hundreds, to combine them to forecast a probability or a value of the desired weather element, termed "the predictand" in statistics. Regression does this by some form of line or curve fitting. The technical form of regression is that the predictand (Y) is fit to a function of the predictors (X): Y ~ f(X,Beta), where "Beta" depends on the specific regression method used. Neural Networks are another form of curve fitting. Neural Networks may have multiple levels of fitting between the predictors and the predictand, and utilize a sigmoid function as the curve to fit (Figure 2). Model Output Statistics (MOS) is utilized by the National Weather Service to forecast many surface and near-surface weather parameters. MOS utilizes multiple linear regression based on multiple years of historical model data and corresponding weather observations. MOS does quite well forecasting surface temperature, dewpoint and other surface values, which have a smooth continuum of values, and have nearly linear responses to their predictors. For non-surface challenges such as precipitation or thunderstorms, regression or curve-fitting methods can suffer from algebraic instabilities, and assumptions on the nature of the data (for example, errors should have a Gaussian distribution) are less valid.

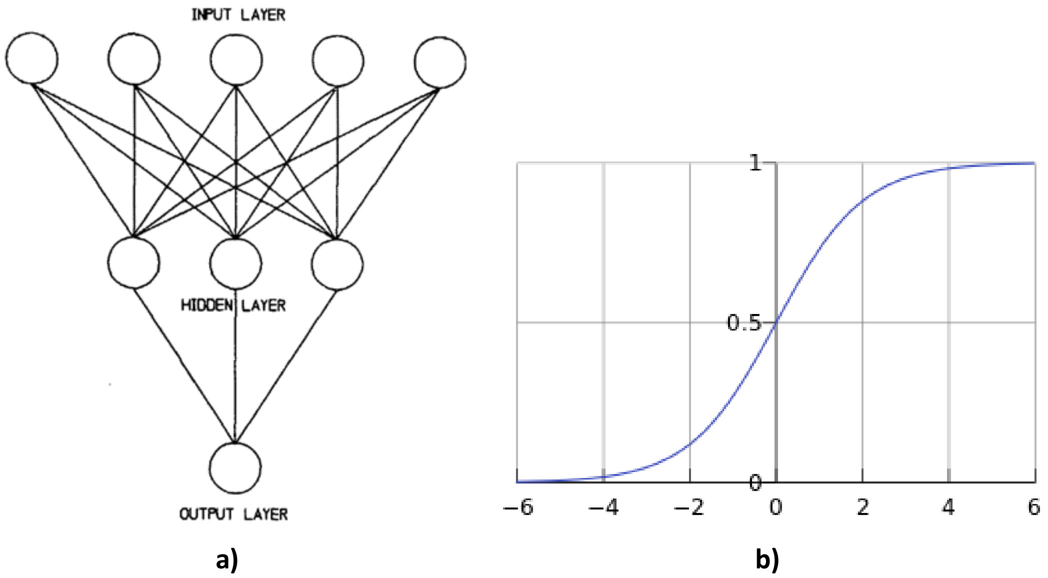


Figure 2. a) Schematic illustration of a neural network (from Figure 3, McCann 1992). Predictors (input layer) of the predictand (output layer) go through two levels of weighting. Lines illustrate the potential for every predictor to have non-linear interactions with other predictors. b) Sigmoid curve used to fit predictors to predictand(s) in a neural network.

Similarly, data mining methodologies used in weather forecasting also have the goal of relating combinations of predictors to the target weather parameter (predictand). In place of curve fitting, most data mining methods have so far relied on some type of data structure as the vehicle to fit predictors to predictands. Decision trees, clustering, and contingency tables (Figures 3, 4, and 5) are examples of data mining methodologies. Decision trees generally use a strategy to repeatedly divide data into a favorable event branch and an unfavorable non-event branch, repeating the division of data as needed.

Breiman 2001 recognized drawbacks of decision trees, and created the Random Forest (RF) methodology to address those weaknesses. One of the weaknesses of decision trees is that they split data into two portions, splitting a dataset *where it is most* sensitive to the predictand, that is, where small change in a predictor value corresponds to larges changes in the predictand. Subsequent splitting of the decision tree process provides the opportunity to refine a reduced portion of the dataset. Breiman used a powerful methodology for addressing this weakness. Using random numbers, 500 different decision trees were created from approximately 2/3 of the historical data record. From this ensemble of decision trees (the "forest"), a consensus or average was derived as an improvement of the "straightforward" decision tree process.

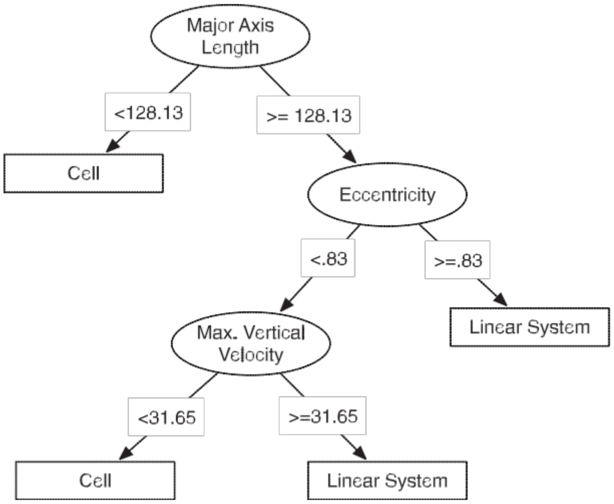


Figure 3. An example of a decision tree, used to distinguish between linear and cellular radar echoes, based on radar image data. From Figure 1, Gagne et al. 2009.

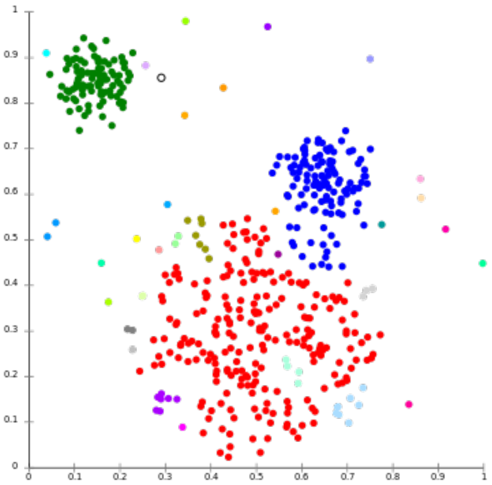


Figure 4. Clustering in two dimensions. Colors indicate data points in the same cluster, hopefully representing the same kind of physical object. From "Cluster Analysis", Wikipedia.org.

Contingency tables are another form of data mining. A table or tables with probabilities of a weather parameter from historical data can easily be created and used to forecast future events. A difficulty with contingency tables is that beyond three or four predictors, it is extremely difficult to fill a 3 or 4 dimensional table, as there is not enough data. An example of a successful four dimensional table was the Air Force Weather Agency (AFWA) Stochastic Cloud

Forecast Model, which had 10 categories of pressure (i.e., the vertical height coordinate), temperature, relative humidity, and vertical velocity, which were used to forecast cloud amount. McDonald (personal communication, 2014) noted that the vertical velocity bins had to be handled with care, as the distribution of vertical velocity was not very uniform. In the experience of the author, success of contingency tables can depend heavily on the "bin size" of the predictors, that is, the range of a predictor value that falls in the same category. One means of handling more than four predictors is to "chain" several contingency tables that are only two dimensional (two predictors), as was done by Keller 1982. Keller has since recognized that this process can dilute information, in that the chaining of contingency tables can lose information about the earlier combinations of predictors. It will be seen that the RNN method attempts to address this limitation of contingency tables by bypassing the creation of a data structure.

		Shear term						
		-77	**0.38**	**-0.2**	**0.2**	**55**	**105**	
Showalter Index	**-0.7**	0	0.193	0.415	0.39	0.468	0.297	0.225
	1.8	0.037	0.165	0.161	0.162	0.317	0.412	0.206
	3.8	0.118	0.106	0.185	0.153	0.228	0.364	0.138
	6	0.068	0.072	0.097	0.107	0.146	0.188	0.139
	8.5	0	0.013	0.096	0.073	0.296	0.252	0.062
	12.5	0	0.024	0.038	0.093	0.079	0.08	0.087
		0	0.006	0.006	0.038	0.066	0.063	0.072

		500 mb wind speed--meters/second						
		8	11	140	17.5	22	26.5	
850 mb temp advection--degr/8 hours	**-1.8**	0.227	0.25	0.142	0.122	0.038	0.013	0.006
	-0.6	0.074	0.086	0.098	0.157	0.079	0.009	0.026
	0	0.082	0.101	0.071	0.261	0.127	0.008	0.04
	0.3	0.072	0.117	0.143	0.236	0.172	0.071	0.104
	0.9	0.037	0.135	0.162	0.304	0.183	0.117	0.065
	2	0.054	0.125	0.183	0.188	0.336	0.196	0.256
		0.019	0.089	0.251	0.36	0.43	0.282	0.325

		300 mb wind speed--meters/second						
		13.2	18.2	22.5	27	33.5	41.5	
850 mb wind speed	4.5	0.061	0.134	0.148	0.113	0.064	0.055	0.019
	6.4	0.039	0.116	0.109	0.155	0.104	0.052	0.019
	8.5	0.093	0.126	0.099	0.178	0.172	0.047	0.026
	10.8	0.172	0.321	0.194	0.181	0.208	0.073	0.097
	13.3	0.145	0.33	0.258	0.28	0.27	0.256	0.142
	17	0	0.416	0.435	0.372	0.215	0.38	0.186
		0	1	0.454	0.258	0.108	0.132	0.087

		500 mb temp advection--degr/8 hours						
		-1.8	-0.8	-0.2	0.2	0.7	1.7	
Showalter change degr/8 hours	-1.8	0.234	0.407	0.449	0.214	0.222	0.559	0.647
	-0.8	0.175	0.316	0.324	0.145	0.262	0.442	0.435
	-0.2	0.114	0.207	0.286	0.144	0.142	0.175	0.367
	0.2	0.105	0.097	0.095	0.085	0.13	0.185	0.189
	0.7	0.096	0.145	0.113	0.05	0.115	0.166	0.122
	1.7	0.105	0.083	0.047	0.25	0.134	0.222	0.118
		0	0	0.25	0.208	0.23	0.25	0.109

Figure 5. Severe weather probability for given combinations of two predictors; i.e., two-dimensional contingency tables. From Keller 1982.

Data mining methods are appealing in that they optimally provide a forecast of a predictand (the weather element that one desires to forecast) based on previously observed values of predictors. However, their ability to do so may be limited, either because there is a data structure involved, or because the amount of data required can be much more than is feasibly obtainable.

In theory, the Nearest Neighbor (NN, not to be confused with neural networks, which will not be mentioned beyond this point) approach is the "best" data-mining methodology, as it utilizes no data structure or curve fitting, thus eliminating shortcomings that might come from those methods. Unfortunately, a straightforward implementation of Nearest Neighbor is difficult to implement, again because of the possibility of needing to match dozens of possible predictors and dozens of predictor bins, resulting in perhaps billions of combinations of data, as explained by Wu et al. 2008, chapter 8.

The NN method is a method of data mining that requires, at first glance, more data than can be gathered. If a prediction problem has 10 predictors, and if the predictor values are divided into a relatively modest (in the opinion of the author) 10 bins, one requires 10^5, or 100,000 data points to store each combination. Practically speaking, there should be at least 100 or 1000 samples of each combination (depending on how much the data points are related or independent of each other), therefore the number of gridpoints required becomes 10 to 100 million. This, the "straightforward NN" approach, makes two naïve assumptions: 1) that all combinations of all predictors will occur with equal frequency, and 2) that all predictor combinations are equally relevant. In practice, gridpoints with turbulence are much more influential in creating predictor-predictand relationships. This paper will demonstrate an approach to NN that does not require millions or billions of predictor combinations to be stored.

Inspired by the use of random numbers in the RF methodology, RNN was designed to be able to implement the NN algorithm effectively. A major goal of RNN was to extract information from historical data in a very direct way, without using any form of curve fitting or data structures (contingency tables, clusters, or decision trees). Section 2 will describe the weather data used in this report. Details of RNN will be given in Section 3. Section 4 will test the ability of RNN to forecast by performing experiments on real and synthetic data. Discussion of these RNN experiments follow in Section 5. Section 6 is a summary of key points.

2. Data

In this report, the intent is to forecast low-level turbulence that occurs outside of convection. Predictors of turbulence were collected from the AFWA WRF 15km model, and corresponding PIREPS (the predictand) were collected between 2013 May 18 and 2014 February 5. At AFWA, CONUS runs of the 15km WRF are at 06 and 18UTC. Since most PIREPS occur between 12UTC and 03UTC, forecasts projections valid at those times of the day were archived. Table 2 lists the model run and valid times. Table 3 lists the model predictors of low-level turbulence that were collected. For this report, only four of the available predictors were used to emphasize the ability of RNN to extract relationships between turbulence predictors and turbulence. The four predictors are boundary layer wind speed, boundary layer lapse rate, a wind shear parameter, and a mountain wave parameterization. Smoothers were applied to the wind shear parameter and the mountain wave term, as graphs of the individual turbulence response with the smoothed predictors yielded a much better discrimination between low and high values of turbulence than without the smoother.

Table 2. Model run and valid times used in this study.

06UTC run		**18UTC run**	
Fcst hr	Valid hour	Fcst hr	Valid hour
6	12	18	12
9	15	21	15
12	18	24	18
15	21	27	21
18	0	30	0
21	3	33	3

Table 3. Model predictors collected for forecasting low-level turbulence. Asterisks indicate predictors used for the RNN study.

BL Wind Speed *	Average of 2 levels: approx 0 and 1600m
Elevation	From the 15km gridpoint
BL Lapse rate *	Lapse rate between approx 0 and 1600m
Max wshear parm	Wind speed changes with height, max value capped
Max wshear parm lgt smth	Same, with light smoother, max value capped
Max wshear parm no cap	Wind speed changes w/ height, max value not capped
Max wshear lgt smth no cap *	Wind spd changes /w height, light smoother, no cap
Mountain wave area	Wind downslope over terrain
Mountain wave area lgt smth *	Wind downslope, light smoother
Panofsky index	Panofsky index for low-level turbulence
Richardson	Using 0 and 1600 feet as the layer
Richardson	Maximum in the layer
Richardson	Minimum in the layer
Valid 8-digit date	Valid time: 2 digit year, month, day, hour
Valid hour	0, 3, 12, 15, 18, or 21

PIREPS, occurring between 0 and 3,000 meters above ground level were put into the AFWA WRF 15km model gridpoint format. The gridpoint nearest to a light, moderate, and severe turbulence report was assigned a value of 1, 2, and 3 respectively. A heavy smoother was then applied with the intention of increasing the areal coverage of the turbulence report, as it is believed that the report is representative of an area larger than a single 15km model gridpoint. The predictand is therefore neither a probability of turbulence nor the intensity of turbulence observed, but a combination of both. For this report it will be termed the “amount” of turbulence. The author believes that the “amount” is more like a probability than an intensity of turbulence. Figure 6 shows an example of PIREPS expanded and added in this way.

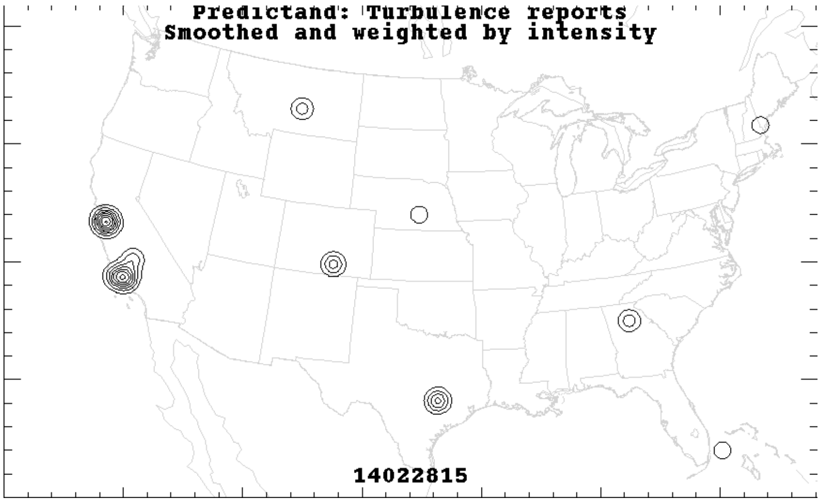

Predictand: Turbulence reports
Smoothed and weighted by intensity
14022815

Figure 6. Turbulence indicated by PIREPS, 2014, February 28, within 1.5 hours of 15UTC. Contours reflect both turbulence intensity and multiple reports, with a heavy smoothing operator.

To properly develop and test a statistical relationship, a historical dataset should be divided into development and test datasets. The most basic reasons for this are to test 1) the statistical stability and 2) the long-term reliability of the predictor-predictand relationships. Discussion of this subject is continued in Appendix A. For this study, half of the data was development data, with odd-numbered valid dates, and half of the data was the test data, with even-numbered valid dates. An oddity of this dataset is that verification data essentially is duplicated. Referring to Table 2, it can be seen that the 18 UTC run and the following day's 06UTC run both forecast for the period 12UTC to 03UTC on the same day. The verification data was therefore duplicated, while the forecasts model data was likely similar but not exactly the same.

The archived dataset is therefore a collection of predictors and the single predictand, and since all data is from model gridpoints or converted to model gridpoints, data elements will be referred to as “gridpoints” throughout this report.

Several quality control measures were taken to insure the integrity of the PIREP data. Gridpoints that were beyond the CONUS land boundary (shown in Figure 7) were not used, assuming that PIREP density was not as good over the oceans. PIREPS were quality controlled in several ways to insure that a valid data file was received. One filter requires that at least one turbulence report is found over the CONUS; this occasionally discarded a valid case with smooth flying over the entire CONUS.

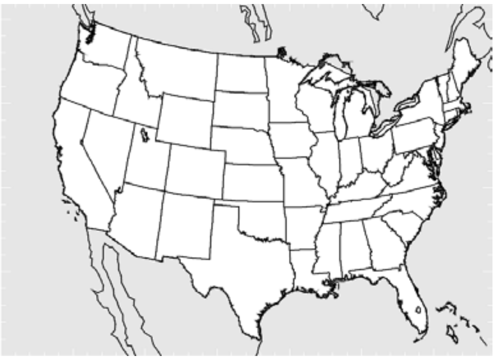


Figure 7. "CONUS land area", white.

Predictors were "normalized" as follows for utilization by RNN data mining. Each predictor was sorted by its value, and divided into 256 bins, each bin having the same number of occurrences. This has several advantages:

1. The predictor is "normalized" in some sense
2. Dividing into 256 bins allows the data to be stored as byte values, which is computationally faster and takes less computer memory
3. The predictor units are irrelevant to the software and data mining
4. The number of bins, 256, is assumed to be sufficient to represent the distribution of predictor values and the resulting amount of turbulence, to extract all available information content from the model predictor data. That is, 256 bins is believed to allow small changes in predictor values to reflect changes in turbulence probabilities, should this situation exist in the real world. As an example, the spike of turbulence occurrence on the high end of the lapse rate predictor "lapsegd", in Appendix B, occurs in the last 10-15 bins. Fewer bins would risk not resolving this spike-shaped response.

Typically in regression forecasting a predictor will be normalized in terms of standard deviations from the mean. Normalization in this way assumes that the predictor data and the probability of turbulence follows a Gaussian curve. The RNN division into 256 bins makes no assumptions about the distribution of either the predictor data or the predictand.

A final filtering process was performed on the historical data. To reduce the amount of data considered, 90% of the gridpoints without observed turbulence were discarded. This was done for the following reasons:

1. Reduce the amount of data to sort and sift with data mining, that is, less data was a time saver.
2. It is believed that the turbulence gridpoints have more statistical influence, i.e., relevance, on the predictor-predictand relationships than the non-turbulent gridpoints
3. It is believed that the non-turbulent gridpoints are largely redundant, but not completely so.

This dataset is therefore biased toward turbulence. To make a forecast in the real world, RNN will have to be run on the complete dataset. However, it is not believed that the bias towards turbulence influences the effectiveness of RNN, or the relationships between predictor combinations and turbulence.

3. Methodology and implementation of RNN

a. Overview of RNN

The goal in developing RNN was to implement the Nearest Neighbor (NN) approach in a straightforward manner. The RNN methodology is described in this section at a high level. Details, technical and statistical notes, and implementation tips will be deferred to Appendix C.

Applied to turbulence forecasting at AFWA, the goal was to look up past amounts of turbulence, given "similar" conditions. As stated above, if all possible predictor combinations are listed, implementation of NN becomes an impossibly large task, due to the extremely large number of possible predictor combinations, which can be a larger number than the actual data that is collected. The synthetic data example in section 4 will illustrate this point.

For this study, the number of predictors used to forecast low-level turbulence was not allowed to vary. The number of predictors used in a regression or data mining process can obviously affect the results. In order to limit this study to the understanding of RNN and its potential for information extraction, the number of predictors was not allowed to vary. In addition, four *specific* predictors were chosen and not allowed to vary, again for the purpose of exploring the capability of RNN as a data mining tool.

A simple description of the RNN process follows. A random gridpoint is chosen from the historical archive. The four predictor values, and the resulting amount of turbulence, were noted at the gridpoint. Gridpoints having values "similar" to the four predictors were found. This group of gridpoints forms a "neighborhood", a neighborhood in a four-predictor space. From the remaining gridpoints, another random gridpoint is selected (i.e., "random without replacement"), and gridpoints similar to the new gridpoint are grouped into a new neighborhood.

Information about the neighborhoods is stored: the predictors used, the predictor values (low and high range), the statistical significance as measured by the Student's T-value, and the amount of turbulence that should be forecast. The "amount" of turbulence was the mean value of turbulence; other choices could have been used. Note that the Student's T-*value* is used, not the probability that a population is different. The Student's T- *value* is open-ended (not limited between 0 and 100%), and for that reason worked better in this study. The neighborhoods are sorted according to their statistical significance. Table 4 illustrates a listing of some sample neighborhoods using the real data of this study.

Table 4. Sample neighborhoods from RNN using real data.

a) Top two RNN neighborhoods according to Student's T-Value

Predictor	Lowest	Highest	T-value	Turbulence	#gdpts
Dynamic	232	255	81.91	0.0056	11084
LapseRate	83	106			
Wshrparam	232	255			
Mtnwave	232	255			
Dynamic	232	255	79.51	0.008	4088
LapseRate	232	255			
Wshrparam	226	249			
Mtnwave	232	255			

b) "Best" neighborhood with largest *negative* Student's T-value, indicating, with high statistical significance, turbulence amounts below the mean (climatology).

Predictor	Lowest	Highest	T-value	Turbulence	#gdpts
Dynamic	0	23	-15.91	0.0011	17309
LapseRate	231	254			
Wshrparam	0	23			
Mtnwave	65	88			

To forecast with current data, apply the sorted neighborhoods in order of their Student's T-value, look up the amount of turbulence that should be forecast, and apply that value to matching gridpoints in the "current" dataset. If a gridpoint belongs to more than one neighborhood, the one with the best statistical significance is used.

In this implementation, RNN neighborhoods were formed using the development portion of the historical data. The neighborhoods can then be used to "forecast" both the development data (which is not "fair"), and the test data. The test data generally will fit slightly less well than the development data, given that the data is adequate and the data mining implementation goes well.

A key design aspect of RNN is that it samples the dataset in order to determine statistical neighborhoods. This is more efficient than a straightforward NN approach of pre-assigning all possible combinations of predictor values and combinations of predictors. Sampling the data allows for a much reduced and tractable number of predictor combinations, i.e., neighborhoods.

The probability of turbulence for each neighborhood is retrieved from this reasonable number of combinations.

In the examples shown in Table 4, taken from the real turbulence dataset, the “best” RNN combination of all four predictors is when the highest values of the dynamic, wind shear parameter, and mountain wave parameter are achieved (bins 232 to 255), but only modest values of the lapse rate (bins 82 to 106). The amount of turbulence forecast is .0056. The next neighborhood states that all four predictors should be at or near their maximum values.

Note that the Student’s T-value is *below zero* when the probability of turbulence is less than climatology, which is 0.0020 in this dataset. Therefore, the *absolute value* of the Student’s T-value was used to rank neighborhoods as the proper indicator of statistical significance.

In Table 4b, the lowest Student’s T-value is -15.91, with a turbulence amount forecast of .0011. This is achieved with low values of the dynamic and wind shear parameters, a modest value of the mountain wave parameter, and a high value of the lapse rate. As shown by the large (negative) T-value, this is a statistically significant neighborhood, with a large number of gridpoints (17309). With a high amount of confidence, one can forecast a low amount of turbulence. In this case, the forecast turbulence will be low in spite of the spite of the large value of the lapse rate.

A key variable in the RNN process is in choosing a bin size for the predictors, which in turn determines the number of gridpoints in a neighborhood. A small bin size is advantageous in that it allows for good resolution of situations where small changes of a predictor value results in a large change of the predictand. However, a bin size that is too small will have fewer gridpoints per bin, therefore less statistical significance, and may result in wildly varying forecast values. With a large bin size, valid details may be lost. For example, the lapse rate predictor “lapsegd”, in Appendix B, has a significant increase in turbulence at the largest “lapsegd” values. It appears that a bin size smaller than 15 is necessary to fully resolve this feature. So far in the RNN design, the bin size will be the same for all predictors. Future versions of RNN might allow different bin sizes for different predictors.

b. Comparison to RF

A brief comparison is now made of the RNN process to the Random Forest (RF) forecast method, with a short explanation of RF. RF recognizes multiple drawbacks of the use of decision trees in forecasting using multiple predictors. Relevant to RNN, one facet of decision trees is that they subdivide datasets using hard threshold values. As an example, if the Lifted Index (LI) is used to forecast lightning, conventional wisdom states that negative (positive) values of LI are associated with lightning (no lightning). A decision tree might divide the dataset near zero, with values of e.g. -0.5 going into the “lightning” group, and values of +0.5 going into the “no lightning” group. One can see that it might instead be desirable to place values of -0.5 and +0.5 into the same category, one that is neither “for” or “against” lightning. The decision tree will attempt to mitigate the initial mismatch at a lower level tree split. Data from Venzke 2001 illustrating this point is presented in Appendix D.

A novel approach used by RF to mitigating decision tree issues is the use of random numbers to choose predictors and predictor combinations, with decision trees as a vehicle. The use of random numbers mitigates drawbacks from the use of hard threshold values. An important aspect of RF is that it creates a large number of decision trees. The number of trees is not fixed but may be on the order of 500 different decision trees, each of them different as random permutations of predictors are used to produce them. The term Random Forest obviously refers to a large number of semi-randomly generated decision trees. To make a real-time or operational forecast, data is run through the 500 decision trees. From the 500 decision trees, some type of consensus forecast is used for the “final” forecast.

The RF process, an ensemble of permutations of decision trees, is quite analogous to ensemble modeling, known to the meteorology community.

Whereas RNN has “neighborhoods”, RF has “nodes”. RNN neighborhoods are a group of gridpoints with predictors having “similar” values. The analogous RF features, tree nodes (leaves of a decision tree), are also a group of gridpoints with predictors having similar values. In Figure 3, the “cell” and “linear system” tree nodes are analogous to RNN neighborhoods.

A gridpoint in the RF technique will have been assigned to dozens, if not hundreds of predictor combinations, and an ensemble of them may have more information than the RNN technique. A drawback may be that a consensus forecast used by RF may instead dilute the forecast toward climatology, due to the possibility of a large range of forecasts. No investigation has been done to support this speculation.

RF has advantages over RNN. A significant feature of RF is that it automatically chooses the number of predictors and predictor combinations; in fact, these can vary enormously from decision tree to decision tree, where RNN has not yet been developed to do this. RF examines a very large number of predictor combinations by repeated sampling of the gridpoints, and choosing different predictors for that gridpoint. In this report, the number of predictors used in RNN has been fixed at four. The purpose of this is to facilitate the understanding of the effectiveness of RNN in “data mining” useful information from predictors of low-level turbulence, not to study an effective predictor selection technique. It is acknowledged that choosing the number of predictors and predictor combinations to utilize is an integral part of all types of human generated, regression, and data mining forecasts. *Predictor selection* techniques are listed in Table 5. RF also has an effective means of dealing with missing predictor data.

This report will not address these issues, but will confine the discussion to the ability and effectiveness of RNN in extracting information.

Table 5. List of commonly used predictor selection techniques in regression and data fitting.

Forward	From all remaining predictors, add the best predictor
Backward	Start with all predictors, remove the least significant predictor
Stepwise	A combination of Forward and Backward, using significance tests
Genetic	Random permutation: adding, subtracting, or changing a predictor

A sample RNN forecast created from data which is *not* a part of the data collection used in the study, is shown in Figure 8.

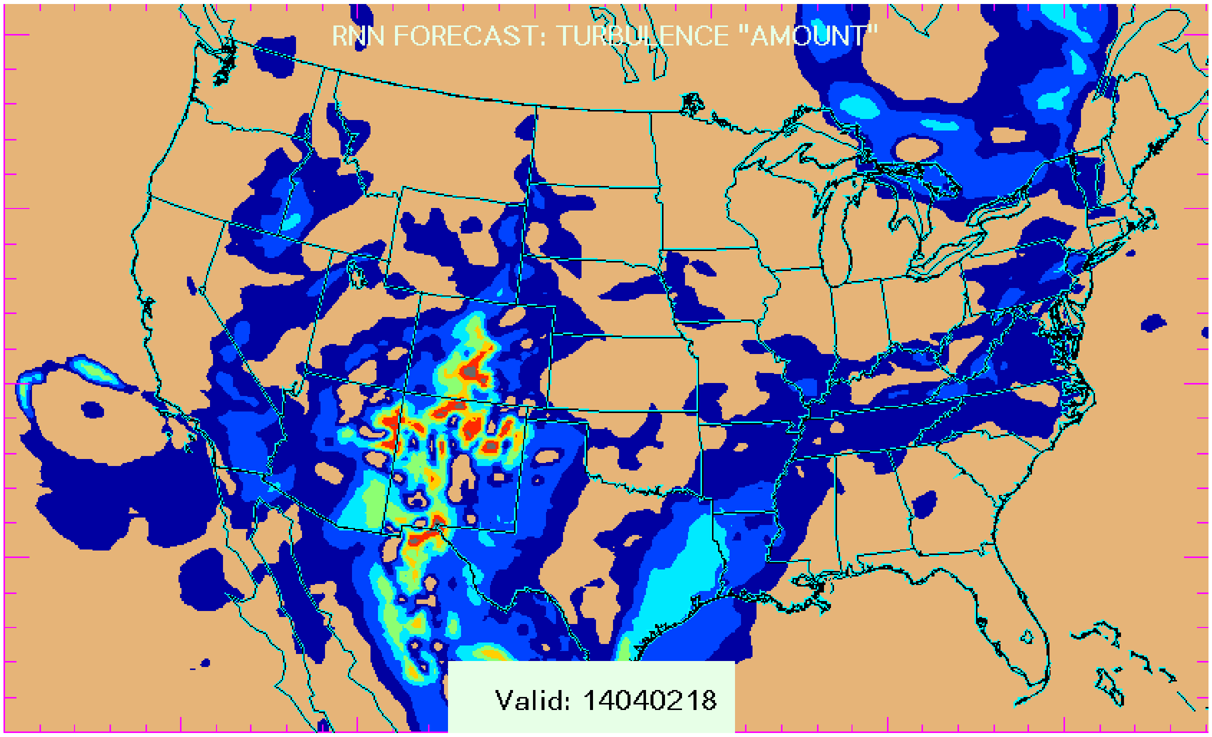


Figure 8. RNN low-level turbulence forecast from 2014 April 2 06UTC run valid 18UTC.

c. Comparison to k-nearest neighbor clustering

Of the "Top 10 algorithms in data mining" listed by Wu et al. 2008, k-nearest neighbor (k-nn) clustering is the algorithm that is most similar to RNN. Like RNN, k-nn also uses random numbers to initialize clusters, analogous to RNN's neighborhoods. K-nn clustering algorithms generally require the need to specify a "distance" metric that specifies how closely a candidate gridpoint matches an existing cluster. The distance metric can be problematic, especially when predictors have different units of measurement. For example, how does one determine how many joules per kilogram are equivalent to per second of wind shear? Also, while adding gridpoints to clusters, the cluster itself changes, and some iteration must be done. RNN in contrast ignores distance metrics and units of measurement by dividing data into 256 bins, all with an equal frequency of occurrence. RNN neighborhoods are “seeded” at random, but do not change as other gridpoints are added. K-nn clustering generally requires a user to decide how many gridpoints are assigned to a cluster; RNN utilizes the Student’s T-value to evaluate the significance (i.e., reliability) of a neighborhood. RNN is intended to be a more direct method of data mining; it is meant to look up, in the most direct manner possible, the value of the predictand from historical values in in an RNN neighborhood of “similar” gridpoints. Finally, RNN seems to be more suited to produce a quantitative result, and k-nn clustering is more naturally suited for classification tasks.

4. RNN Experiments and Notes

a. Linearization of predictors

It is possible to “linearize” predictors of weather events. Linearization transforms piecewise values of a weather predictor to the probability or intensity of the weather event that is expected from a historical dataset.

The advantage of linearizing a predictor is that the resulting forecast will have a better fit to the predictand than a curve fitting technique. The general linearization process is to divide a predictor into a relatively large number of bins, as many bins as the data appears to statistically support, without unnatural variation from bin to bin. For each bin, the amount of turbulence is found by simply looking up the amount of turbulence from historical archives of gridpoints that have the same predictor values in the bin. Graphs of the predictors in this study, linearized, are shown in Figures B1-B15 of Appendix B.

From Tables 1-15 in Appendix B, it can be seen that the correlation of single predictors to linearized turbulence predictors is, in most cases, significantly higher than the predictor in raw form. It can also be seen from the graphs in Appendix B that dividing the turbulence predictor data into 256 categories, a relatively large number with a correspondingly small bin size, appears to be reasonably well supported for the turbulence dataset, evidenced by the smoothness of the graphs. If the response is *not* smooth, one can expand the bin size until consecutive bins have a more consistent, more believable response.

Therefore, an important test for RNN is to see if it will, for a single predictor, duplicate the straightforward linearizing of that predictor. RNN was used with single predictors to forecast low-level turbulence. Table 6 shows that a straightforward linearization of the dynamic predictor, calculated independently of RNN, *has equal skill* to RNN, as the correlation of the linearized predictor with turbulence were identical, to three decimal places. The only algorithmic difference is that RNN picked predictor bins randomly, while the "straightforward" linearization calculated the bins methodically. A bin size setting of "1" was used, which means that each of the predictor's 256 values were used separately to look up the amount of turbulence observed for that predictor value. An oddity is that the test data fit slightly better than the development data; the reverse is normally expected.

Table 6. Linearizing dynamic predictor without and with RNN: correlation coefficient.

Dynamic Predictor	Linearized	RNN
Devt data	0.140	0.140
Test data	0.145	0.145

By extension, it is evident that using two, three, or more predictors with RNN is the equivalent of *linearizing two, three, or more predictors at once.* This will be true if there is adequate data, and if there is actually information contained in multiple-predictor combinations. This will be investigated with further RNN and data mining experiments.

b. RNN with synthetic data

Two synthetic datasets were created to test the limits of the RNN technique. Consistent with the four predictors used for low-level turbulence, four mock predictors, ranging in value from 0.0 to 1.0, were created. As was done with the real dataset, the predictors were normalized to values ranging from 0 to 255 by their frequency of occurrence. The mock response to each predictor by themselves is listed in Table 7.

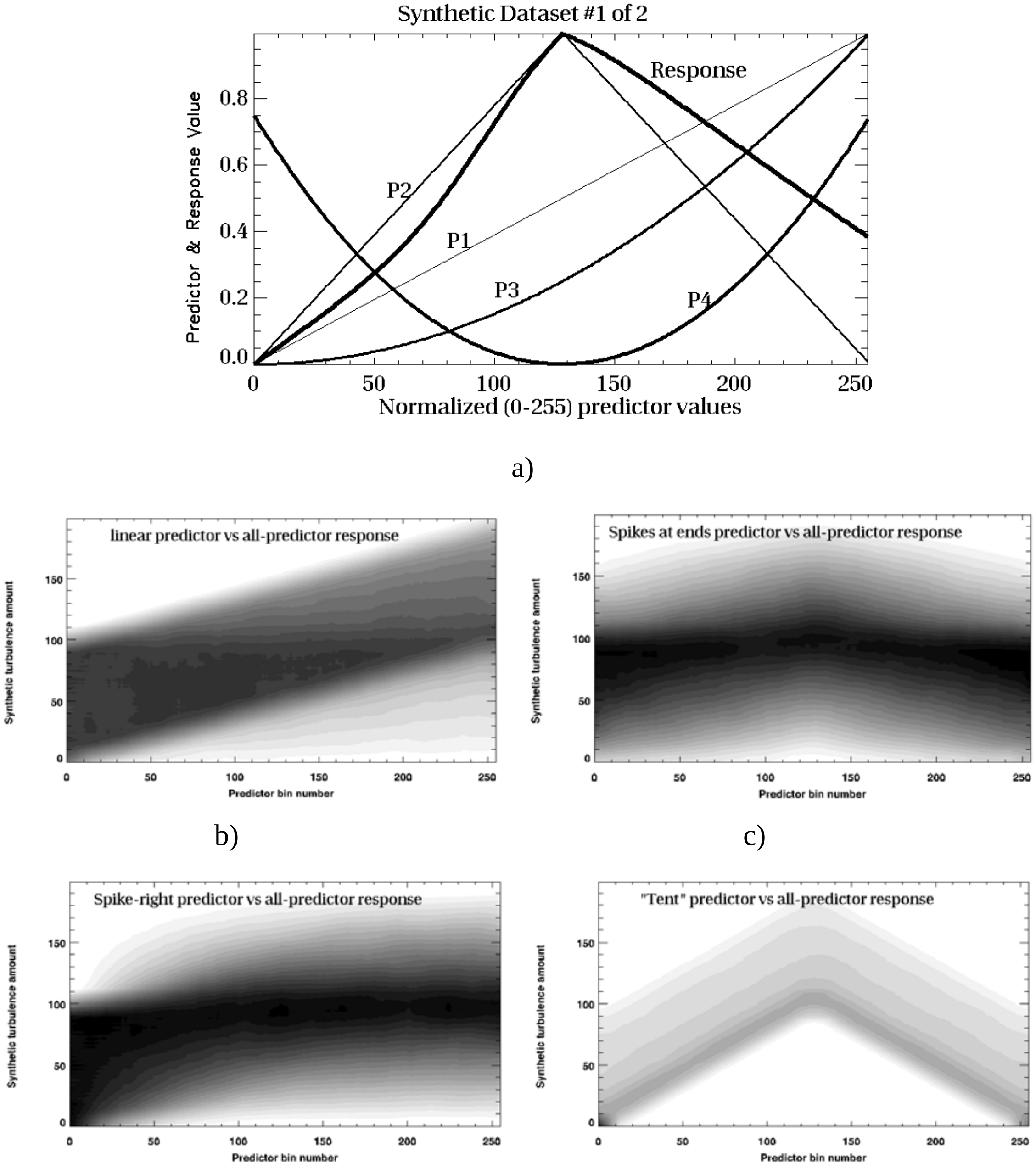
Table 7. Response to synthetic predictors.

Predictor 1	Linear with the predictor ("Linear")
Predictor 2	Tent-shaped peak in the middle ("Tent")
Predictor 3	Spike at the high end ("SpikeRight")
Predictor 4	Spikes at both low and high ends ("SpikeBothEnds")

The mock response to all predictors, in combination, intending to simulate non-linear responses that might exist in a real atmosphere, is given by the equation below. Graphs of the response of each individual predictor and the combined predictor set is shown in Figure 9.

$$\mathbf{Response = Tent + Linear * SpikeRight\ \ / (SpikeRight+SpikeBothEnds)} \qquad (1)$$

Synthetic dataset #1 was created such that predictor a=b=c=d, that is, all predictors have the same values, however, this is not true of the response, which varies. The unnatural aspect of this synthetic dataset is that all of the predictors are “low” or “high” at the same time. This experiment nevertheless demonstrates how RNN works.



a)

b)

c)

d)

e)

Figure 9. a) (Large line graph above.) Synthetic dataset predictand response to individual predictors, predictors have normalized values (0 to 255). b) through e): Combined predictor response, shaded, to synthetic dataset #2, versus single predictors.

Results of the RNN fitting to the development portion of synthetic dataset #1, and applied to the test data, are shown in Table 8a. Another RNN run (not shown) with a smaller bin size yielded a correlation increase to 0.992. Thus, RNN successfully replicates a synthetic dataset with a non-linear, multiple-predictor response function to the predictors, nearly perfectly. The number of RNN neighborhoods was 72 with the larger bin size, and 570 with the smaller bin size.

Table 8. RNN-fitting results to synthetic data.

"SIMPLE" SYNTHETIC DATASET #1			"COMPLEX" SYNTHETIC DATASET #2		
	Unfit gridpoints	Correlation		Unfit gridpoints	Correlation
Dev't data	0	0.987	Dev't data	0	0.987
Test data	0	0.986	Test data	5980	0.977

Since synthetic dataset #1 had an unrealistic predictor set, in which all of the predictors had low/middle/high range values at the same time, synthetic dataset #2 was created. Synthetic dataset #2 has predictor values that are completely random, ranging from 0.0 to 1.0. This is also unrealistic, in that predictors in any real dataset are likely to be correlated. Correlations of the real turbulence data predictors in this study are in Appendix E. Synthetic dataset #1 is unrealistic in that all predictors have a 100% correlation with each other, and synthetic dataset #2 is unrealistic in that the predictors are completely *uncorrelated* with each other (0% correlation). Nevertheless, the synthetic data experiments will demonstrate RNN's potential to extract information from a multiple predictor, single predictand dataset.

For synthetic dataset #2, the number of permutations of 4 predictors with 256 values is 256^4, which is 4,294,967,296 combinations. This is a much larger number of combinations than has been actually collected from the real data of this study. For the experiment, one million random gridpoints with random predictor values were generated. These were divided into a developmental and a test dataset by choosing every other gridpoint. The bin size was set to 25, which was chosen after experimentation to allow the data fitting to run to completion in a reasonable time. A bin size of 25 means that each bin covered about 1/10 of the range of a predictor.

To minimize the possibility of mistakes, the same code was used to run RNN on the synthetic data as was used with the real data used in this report.

The results were impressive. RNN ran without any computational trouble on synthetic dataset #2. 65563 neighborhoods consisting of gridpoints with "similar predictor values" were created by RNN. The developmental data was then fit with the RNN neighborhoods, and finally the test data was forecast with the same neighborhoods. In summary, the "test" data was forecast almost perfectly, achieving nearly a 98% correlation coefficient! A small percentage (5980 of 500,000) of the test gridpoints failed to have forecasts. A backup forecast, climatology, was chosen for those gridpoints.

The number of neighborhoods created, 65563, is believed to be extremely high since the synthetic data predictors are randomly chosen, and therefore predictors are completely uncorrelated. Real world predictor data will likely have appreciable correlation; therefore RNN will select fewer neighborhoods, as more gridpoints will have combinations of predictor with similar values.

It is instructive to study the average number of gridpoints in each neighborhood that is generated by RNN with the synthetic data. One might expect that 500,000 gridpoints (in the development dataset), divided by 65563 neighborhoods, would result in approximately 7.63 gridpoints per neighborhood. However, a gridpoint can be in several neighborhoods. Querying the program, the total number of gridpoints in neighborhoods was 2,475,219. Therefore, it seems that a gridpoint was typically found to belong to about 5 different neighborhoods. This is deemed to be normal behavior for RNN. Recall that when making a real-time forecast, if a gridpoint matches multiple neighborhoods, the "best" neighborhood is applied, according to the chosen metric, the Student's T-value.

c) RNN with 2, 3, and 4 predictors using real turbulence data

The goal of creating RNN was to forecast low-level turbulence by extract the maximum amount of information from predictors of low-level turbulence. RNN was run with two, three, and four predictors. RNN using one predictor has already been done via the linearization experiment. Table 9 shows the RNN results with varying numbers of predictors and bin sizes.

Table 9. RNN results with real data. Best correlation for test dataset while fitting to development data, using 1, 2, 3, and 4 predictors.

#Predictors	Best Correlation	Bin size	# RNN neighborhoods selected
1	0.145	3	871
2	0.154	17	3611
3	0.162	23	6331
4	0.158	43	8397

Forecastability results with 2, 3, and 4 predictors were slightly better than the best single-predictor correlation. The best single predictor was the dynamic term, with a linearize correlation coefficient of .145. The best RNN run came from using three predictors, with an independent data forecast correlation of .162, which is almost 12% better than the single best predictor.

There is evidence that the fourth predictor is degrading the data fitting, as adding that predictor, a simple mountain wave parameterization, failed to increase the correlation from three predictors. Possible reasons for this are discussed in Appendix F. While a strategy to mitigate this should be developed for future RNN projects, by noting the problem and being watchful for over-fitting due to too many predictors, RNN can be successfully implemented. As was seen in the synthetic dataset, there is still a great potential for RNN to accurately fit complex multiple predictors.

d. RNN Notes

Using RNN on real data, the development portion, it was noted that RNN slowed greatly while categorizing development gridpoints into neighborhoods. This occurred when RNN was using three or more predictors. Investigation revealed that gridpoints not fit into neighborhoods within the first 90% of the process are very "uncommon", and forming neighborhoods with the last 10% of the gridpoints is difficult. These gridpoints are ones that have unusual combinations of the values of three or more predictors. Examination of the neighborhood output showed two patterns. It seemed that many of the predictors were at their lowest possible bin value, i.e., normalized predictor value of "0". Also, the predictor combinations near the end of the RNN neighborhood grouping process were predominantly forecasts of low amounts of turbulence. Two possible reasons for this are that the dataset was biased towards forecasts of turbulence, and/or that there are more and disparate combinations of predictors for *low* turbulence amount than for high turbulence amount. That is, one or two predictors *favorable* for turbulence are negated by the remaining predictors being *unfavorable.* An implication of this will be discussed in Section 5.

It should be noted that the process of forming neighborhoods appears to be inherently inefficient in a computational sense. Searching an array for combinations of several predictors, each predictor having a certain range of values, appears to require a large amount of CPU usage.

5. Discussion of the RRN experiments

In applying RNN with multiple predictors of low-level turbulence, only a modest improvement of forecast skill was achieved versus the best single predictor. The correlation coefficient between the test data and PIREPS increased from .145 (one predictor) to .162 (three predictors). Recall the synthetic datasets show that RNN appears to be extracting the maximum possible forecast skill from multiple predictor combinations, even when synthetic dataset #2 was made to be intentionally difficult.

For the low-level turbulence forecasts, possible explanations for a low increase of skill using several predictors over one predictor are:

1. Model data is inadequate to forecast low-level turbulence
2. Inadequacies of PIREPs make data mining difficult without modification of the methodology
3. The data is being divided into bin sizes that are too small to be significant, that is, statistical degrees of freedom have been exceeded
4. The geographical density of PIREPS are inadequate to accurately define an area that should be forecast as turbulent
5. Some of the turbulence predictors are ineffective, therefore diluting the information content of the better predictors.
6. The RNN policy of choosing the single "best" neighborhood, if there are multiple neighborhood choices, is poor.
7. The metric, correlation coefficient, is not a proper choice for the development process when used for turbulence forecasts.
8. The predictand was the *mean* turbulence in an RNN neighborhood. Emphasizing "potential" values of turbulence, such as the maximum turbulence or the maximum 10%, may have different results.
9. The study matches predictors at a model gridpoint with the predictand, but this is an inadequate model of turbulence forecasting.
10. The predictand, a smoothing of turbulent PIREPS, added together, is inadequate
11. Better predictors exist to forecast low-level turbulence.

Reason 4, the adequacy of PIREP data for development of statistical relationships, is believed to be the predominant reason for the low amount of additional skill found with additional predictors. In the subjective opinion of the author, the density and pattern of low-level PIREPS, when plotted on a map simply do not match well with troughs, ridges, wind maxima, or any other weather elements that our models forecast well. An exception, not addressed in this report, is terrain related features, as turbulence is well known to occur frequently in rugged terrain.

Reason 1 suggests that parameters forecast by computer models simply do not have significant relationships to turbulence observed in the atmosphere.

Reason 2 is the admission that a PIREP, or lack of a PIREP, cannot be taken literally as the verification at that point in space and time in the atmosphere. Brown and Young 2000 point out that "Because icing and turbulence observations are not consistently available at the same time and location, pilot reports do not provide a representative sample to the forecast grid". A PIREP with turbulence of any intensity is almost surely an indication that turbulence occurred, but only literally at that minute in time. Lucas 2013 from personal experience notes four aircraft landing

30 seconds apart at an airport can have different amounts of turbulence, ranging from none to moderate. A PIREP of “no turbulence” may therefore only be indicative of no turbulence during that minute in time, and cannot be taken to mean that no turbulence should have been forecast. The implication is that transferring PIREPS directly to a model grid for data mining is inadequate, and that heavy modification of the dataset, based on the reliability, and the space and time scope of the PIREP and the predictor conditions, needs to be taken into account. As an example, one might wish to allow a turbulent PIREP to “verify” turbulence for a few hours before and after the nominal time of the report, and not to penalize a forecast of turbulence as a false alarm during that time period.

Reason 3 concerning small bin sizes is not believed to be significant for the following reason. Recall that the primary variable factor in RNN is the size of a predictor bin. Smaller bin sizes allow for more accurate precision, but are not statistically reliable. The reliability is insured by utilizing the *independent* dataset to select a bin size that produces the highest correlation coefficient. Cross-checking with the so-called independent data is a strong influence on keeping bin sizes large enough to be consistent.

Reason 5 suggests that a predictor with little correspondence to turbulence might act as a random number generator, essentially diluting valid information content of other useful predictors.

Reason 6 is believed to be an insignificant contribution to the ultimate forecast product. While the difference between two competing forecasts was not examined in detail, it was noted that neighborhoods sorted by the Student’s T-Value did have similar forecast values; therefore, it is assumed that the contribution to error is small.

Regarding reason 7, a different metric than the correlation coefficient to measure success might indicate a larger amount of “success”, for example, emphasizing the Probability of Detection (POD) might be desirable in turbulence forecasting. The correlation coefficient is valuable in that it is a metric that is not easily “gamed” by a clever forecaster or by unusual weather conditions. However, it can suffer from “high leverage” data points, since data points that are far from the mean contribute according to the square of their error. In the case of turbulence, nearly every turbulent gridpoint will be a ‘high leverage’ data point, having a relatively large influence on the correlation coefficient. For this reason, other metrics should be used to evaluate the RNN forecast. Brown and Young 2000 cover verification statistics as applied to turbulence. The Joint Working Group on Forecast Verification Research website 2013 lists the pros and cons of different verification scores in meteorology. Finally, if turbulence is actually difficult to forecast, it may be that a forecaster will simply have to choose a desired level of POD, and the forecaster will have to suffer whatever False Alarm Ratio (FAR) that comes with it.

Reason 8: Since PIREPS are believed to have incomplete information about the state of turbulence in the atmosphere, it is felt that different data processing of PIREP data, and different statistical metrics should be used. A value such as the upper quartile, or even the highest 10% of the turbulence amount found in an RNN neighborhood, might be a technique to mitigate the common belief that turbulence is under-reported.

Reason 9 is felt to be a significant issue. The assumption that a turbulent PIREP occurs because of model conditions at the nearest gridpoint can be challenged. Predictors that instead indicate the “worst” condition “in the vicinity” might be helpful. Matching a PIREP precisely in space and time to the corresponding predictor values of the nearest model gridpoint should be relaxed in some way, without causing side-effects in the forecast (such as a higher false alarm rate). Perhaps if turbulence is observed, the data collected should be the “maximum” model predictor value within some distance, and if no turbulence is observed, use the nearest model gridpoint. Also, larger scale “pattern matching” predictors, as opposed to gridpoint indices, might be appropriate for turbulence forecasting.

Reason 10: The predictand data was a smoothed gridpoint, the value of the single gridpoint being proportional to the intensity of the turbulence (1, 2, 3 corresponding to light, moderate, or severe turbulence). This study should be repeated for moderate or greater turbulence, or just severe turbulence. The suggestion is that moderate or greater turbulence might be more predictable, where light turbulence is essentially unpredictable.

Reason 11: Recall that several predictors were collected for this study but not examined; of those, the terrain elevation (and possible derivations such as terrain roughness), the Panofsky index, and the Richardson number, are believed to be effective in forecasting low-level turbulence.

What is seen in this study is that the RNN experiments so far do not add a lot of value to forecasting low-level turbulence beyond that of the best single predictor, given the conditions of this study: multiple predictors and turbulence all taken from the same model gridpoint, valid at the same instant in time.

A meteorological aspect is now considered. The RNN methodology allows for examination of the neighborhood data to understand the nature of low-level turbulence forecasting. One might ask the question: “are there only contributing factors ‘in favor of’ an event, or are there factors that effectively inhibit or even ‘veto’ an event from happening?” Looking at Student’s T-values of the turbulence neighborhoods shows that Student’s T-values can have positive numbers, indicating that a neighborhood forecasts a turbulence amount higher than climatology, or negative, which forecasts an amount of turbulence below climatology. There is an interesting relationship between high and low forecasts of turbulence, as the magnitude of the Student’s T-values for high turbulence forecasts are much larger than the T-value magnitude of lower forecasts of turbulence. One might think of these as favorable and unfavorable regimes for turbulence. Judging just from the magnitudes of the Student's T-values, the "favorable for turbulence" regimes are stronger than "inhibitors" of turbulence. Conversely, there are a large number of RNN neighborhoods which suggest that a mix of favorable and unfavorable predictors result in a lower-than-average amount of turbulence forecast.

6. Summary and conclusions

A new method of data mining was created to extract the maximum amount of forecast capability from several predictors of turbulence. The new method, Random Nearest Neighbor (RNN), was applied to several predictors obtained from AFWA weather model gridpoints, and PIREPS of turbulence observed at the same gridpoint.

Several aspects of RNN strongly suggest that RNN is an effective information extraction paradigm. RNN has no curve fitting or data structure. RNN was inspired by RF, but does not use decision trees as a vehicle for data fitting. Instead, RNN "looks up" the amount of turbulence from archived data in a fairly direct manner. The success of RNN in accurately linearizing a single predictor (against turbulence) is compelling evidence that RNN is effective in extracting information from a historical data collection. By analogy, RNN essentially linearizes a data fit to *several* predictors. Finally, the RNN fit to the complex synthetic dataset is convincing.

Positive aspects of RNN are that it extracts close to the maximum theoretical information from combinations of predictors, that it does so in a direct, understandable manner, it is simple to program (under 1000 lines of Interactive Data Language source code), and with the transformation of predictors to a scale from 0 to 255 according to frequency of occurrence, has no dependence on the physical units and little dependence upon sensitive ranges of predictors.

Drawbacks of RNN are that it does not currently handle varying numbers of predictors and predictor combinations, that it runs more slowly while attempting to categories the last 10% of the neighborhoods, and that it requires several manually run iterations with varying bin sizes in order to optimize the results. It is expected that future work on RNN would be effective in reducing these weaknesses.

The ability of RNN to extract close to the theoretical maximum amount of information from a predictor-predictand dataset implies that the RNN methodology could become an important standard against which the success of other forecasting paradigms could be measured. Given this bold assumption, RNN could be utilized to understand where practical forecasting information exists, and where information is lacking, in a forecasting environment.

The forecasting of low-level turbulence was disappointing. The combination of four turbulence predictors using RNN resulted in only a small improvement over a single predictor. With confidence in RNN as an effective means of information extraction, one must look elsewhere for the difficulty in forecasting low-level turbulence. The primary weakness is believed to be from applying PIREPS as the complete truth over the entire CONUS domain. The author does not believe that PIREPS of turbulence effectively discriminate between turbulent and non-turbulent areas, as it is not known whether gridpoints near turbulent reports should be considered turbulent or not turbulent. Finally, it is believed that the use of other verification metrics for turbulence verification, beyond the correlation coefficient that was emphasized in this study, may be more appropriate for the evaluation of turbulence forecasts.

References

Breiman, L., 2001: Random forests, Machine Learning, **45**, 5–32.

Brown, B.G., and G.S. Young, 2000: Verification of icing and turbulence forecasts: Why some verification statistics can't be computed using PIREPs. *Preprints, 9th Conf. on Aviation, Range, and Aerospace Met.*, Orlando, FL, 11-15 Sept., American Meteorological Society (Boston), 393-398.

Joint Working Group on Forecast Verification Research, 2013: Retrieved 2014 April 4 from URL http://www.cawcr.gov.au/projects/verification/

Keller, D. L., 1982: A statistical severe weather forecasting technique using satellite soundings and radiosonde data, M.S. thesis, Dept of Atmospheric and Oceanic Sciences, University of Wisconsin, 108 pp.

Gagne, D. J., II, A. McGovern, and J. Brotzge, 2009: Classification of convective areas using decision trees. *J. Atmos. Ocean. Tech.*, **26**, 1341-1353.

Lucas, P. F., 2013: Personal communication, Air Force Weather Agency.

McCann, D.W., 1992: A Neural Network Short-Term Forecast of Significant Thunderstorms, *Weather and Forecasting*, **7**, 525-534

McDonald, D. M., 2014: Air Force Weather Agency, personal communication.

Venzke, K. C., 2001: Development of Predictors for Cloud-to-Ground Lightning Activity using Atmospheric Stability Indices, M.S. thesis, AFIT/GM/ENP/01M-8, Air Force Institute of Technology

Wikipedia.org: Cluster Analysis, retrieved 2014 March 29, from URL http://en.wikipedia.org/wiki/Cluster_analysis

Wu, X., et al: Top 10 algorithms in data mining, *Knowledge and Information Systems*, 2008, **14**:1-37

Links to appendices

Appendix A
Predictor selection and information extraction

In this study, the subject of predictor selection methodology was ignored. It is likely that the subject must be addressed with future work on RNN. Of particular concern is that in this study, using four predictors had slightly less skill than three predictors.

Predictor selection is a strategy for selecting and combining several, perhaps dozens of predictors, in a regression or data mining task. There is no simple, clear rule of thumb for the number of predictors that should be used in such a task. Multiple linear regression and other regression techniques have commonly used a “forward predictor selection” technique (adds the best predictor after trying every available predictor), a “backward predictor selection” (starting with all available predictors, deleting the least effective one), and the “stepwise predictor selection” (a combination of forward and backward, with predictors being added or deleted according to a metric).

RF is perhaps unique among regression and data mining methods, in that it automatically tries all (or a very large number) of predictor combinations and all values.

RNN should be refined to utilize a varying number of predictors and predictor combinations. A strategy needs to be developed that either fits into the RNN paradigm, or perhaps modifies a portion of it.

RNN utilizes the test data in order to calibrate the bin size. This is a bit unusual, as doing so in some sense violates the intention of the test data to be completely independent of the developmental data. Also, in this study, the development and the test data were created from the even and odd numbered days. It would be a “fairer” test, in that it would be more realistic, to have the development and test data consisting of longer periods of data on the order of months. Some care is needed to account for the seasonality of data. For example, if RNN were to be implemented operationally, developmental data might be used from months 1, 2, 4, 5, 7, 8, 10 and 11, with test data from months 3, 6, 9, and 12. This would be a realistic test of RNN, in that the development and test data are appreciably independent of each other, and there is ample opportunity for unusual months or seasons to create realistic difficulties for the forecast system as a whole.

Appendix B

Linearization of predictors of low-level turbulence

Figures B1 through B15: graphs of the turbulence response (y axis) of individual predictor values, normalized by frequency of occurrence, to the range 0 to 255 (x axis). See text for details on linearization and turbulence amount.

After the graphs, Table B1 lists correlation of predictor to predictand.

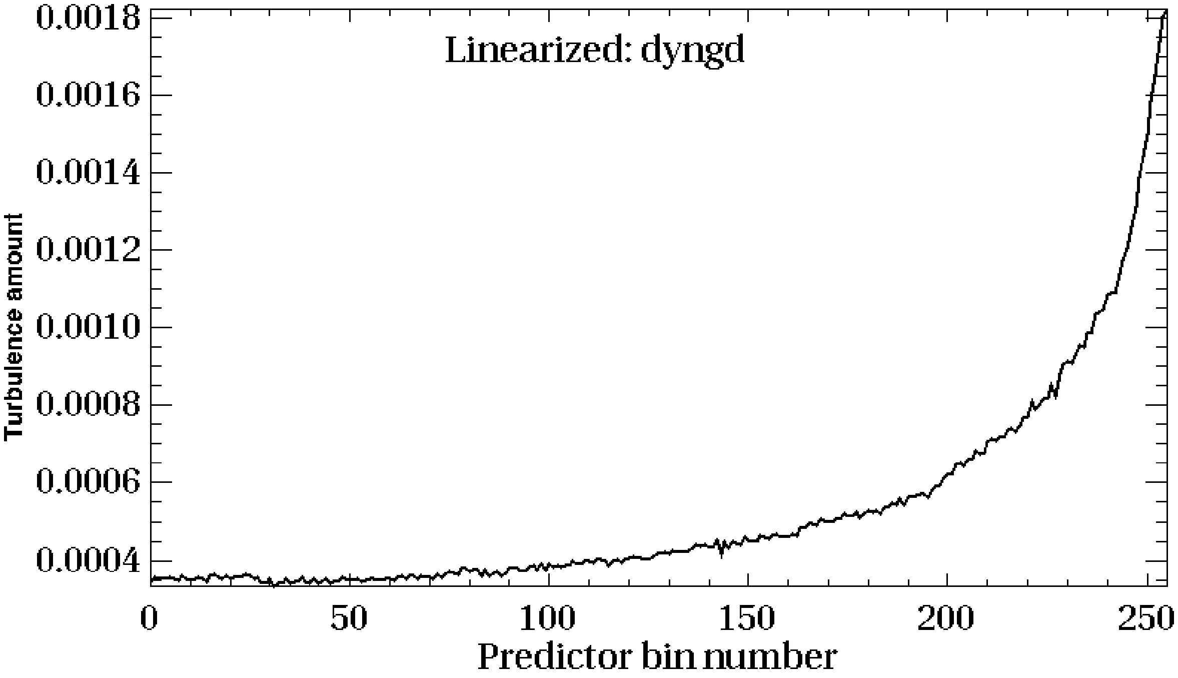


Figure B1. Dynamic (wind speed) predictor; turbulence response.

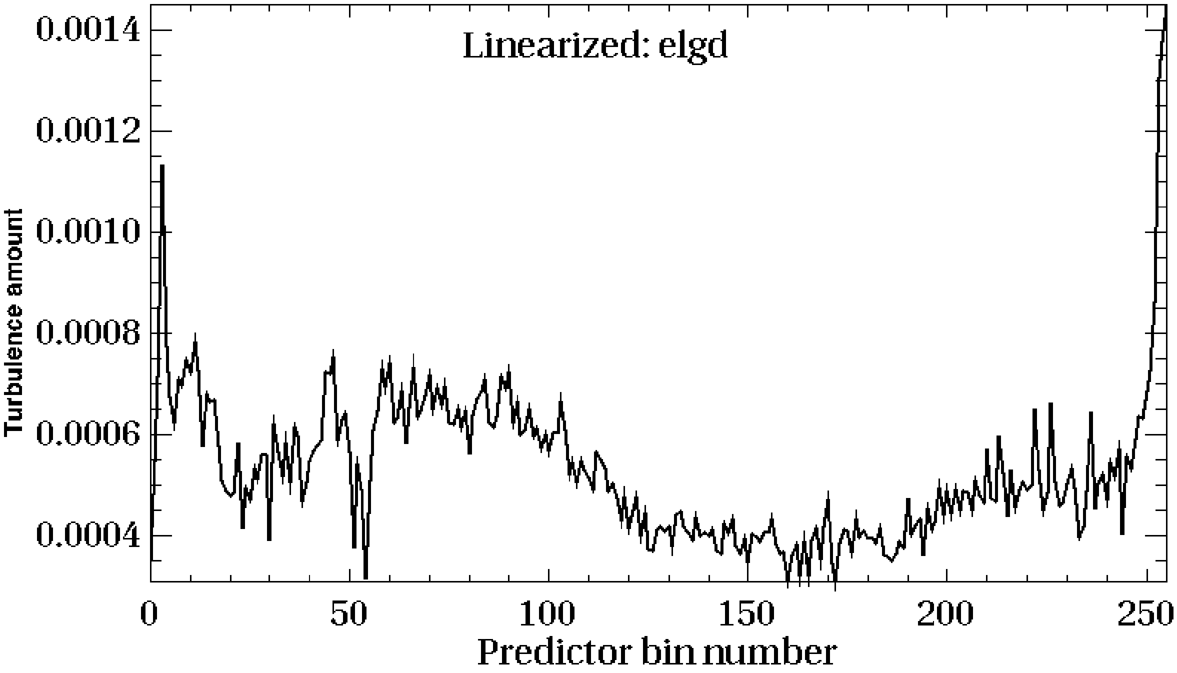


Figure B2. (Model) terrain height; turbulence response.

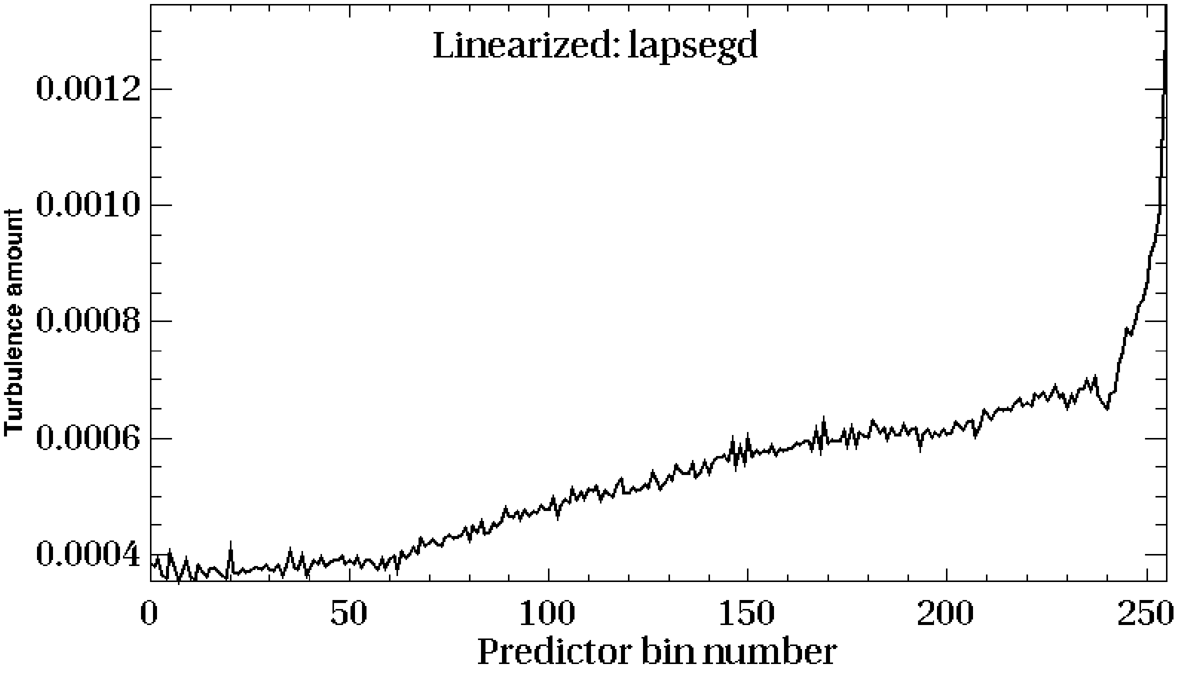


Figure B3. Lapse rate predictor; turbulence response.

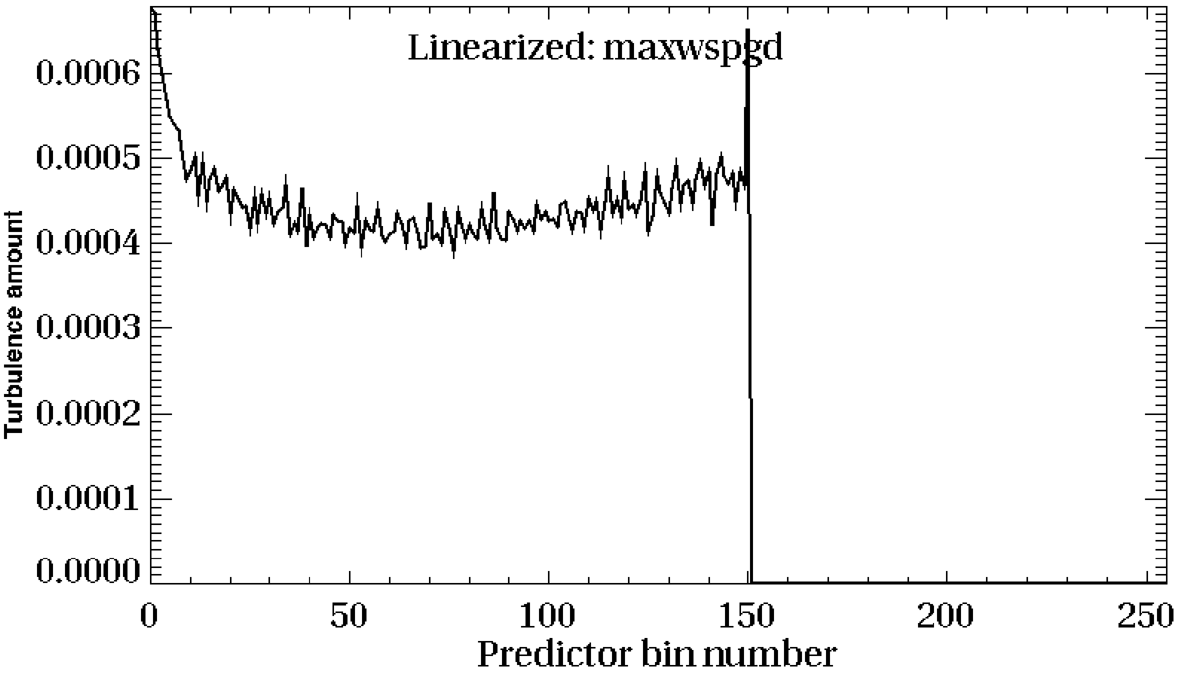


Figure B4. Wind Shear parameter predictor; turbulence response.

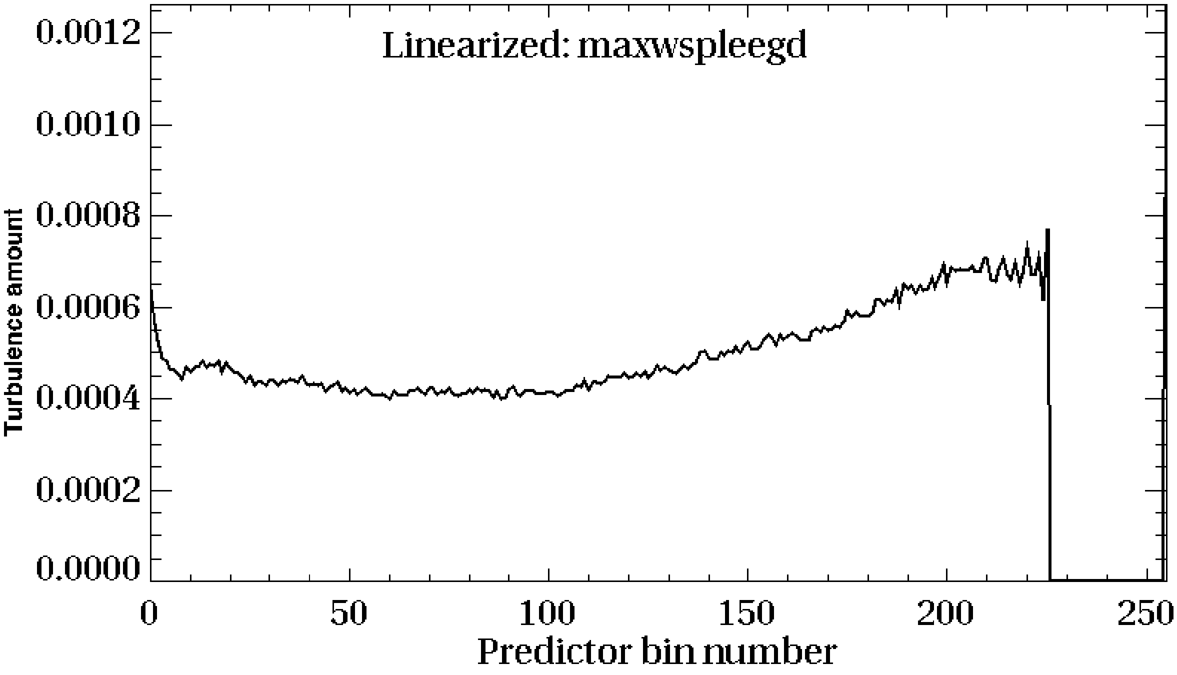


Figure B5. Wind shear parameter predictor, Lee filter applied in 2-dimensions; turbulence response.

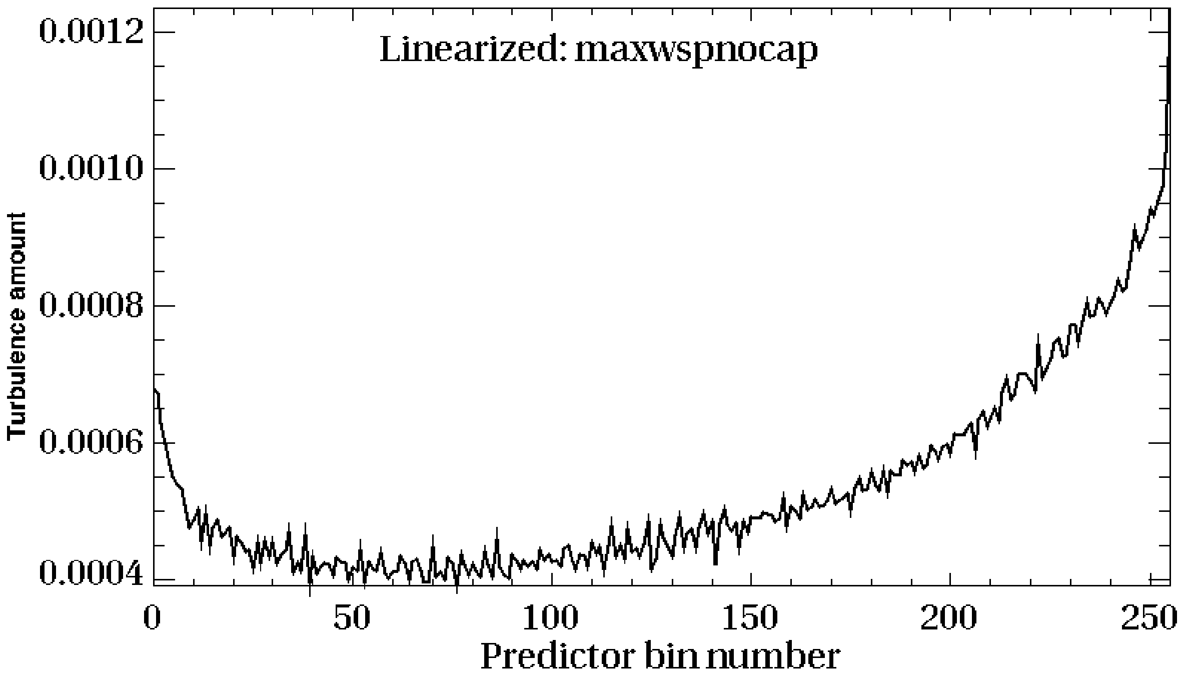


Figure B6. Wind shear parameter predictor, value not capped (limited); turbulence response.

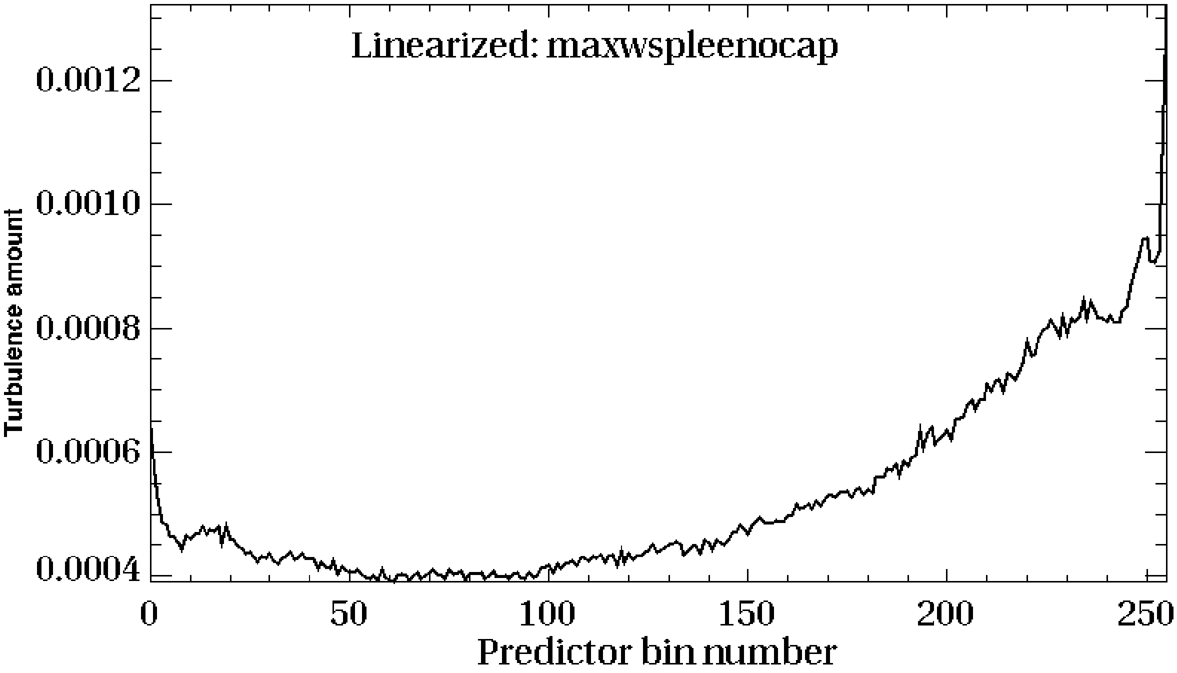


Figure B7. Wind shear parameter predictor, not capped and Lee filter smoother; turbulence response.

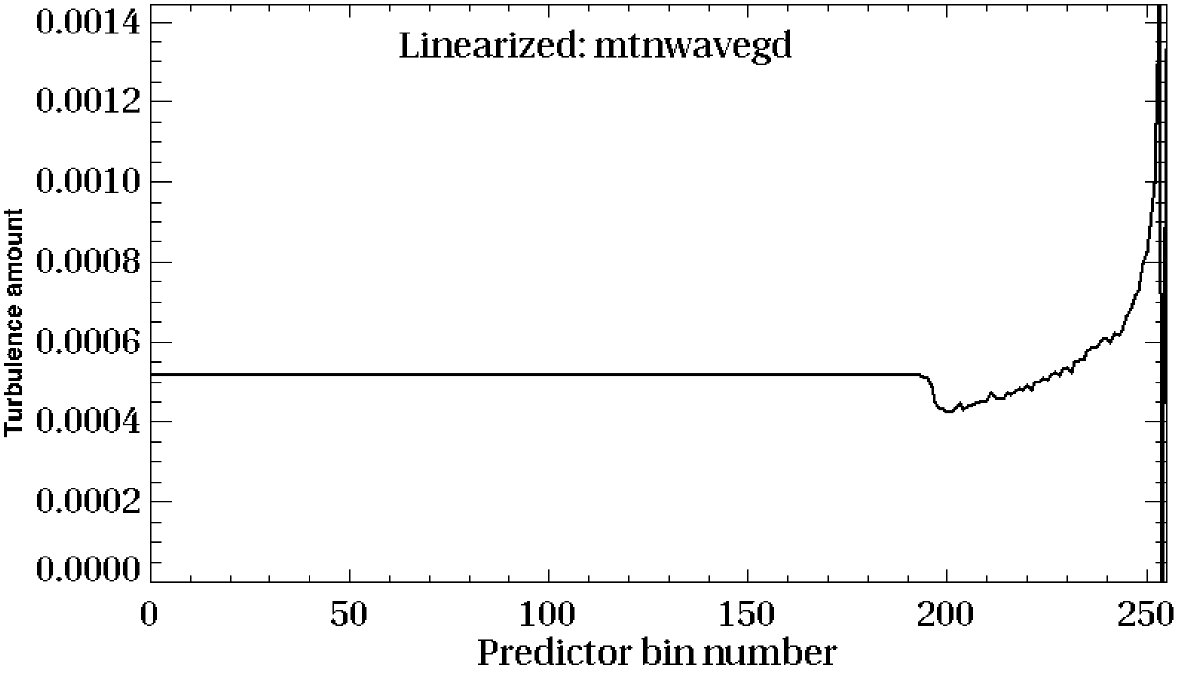


Figure B8. Mountain wave parameter; turbulence response.

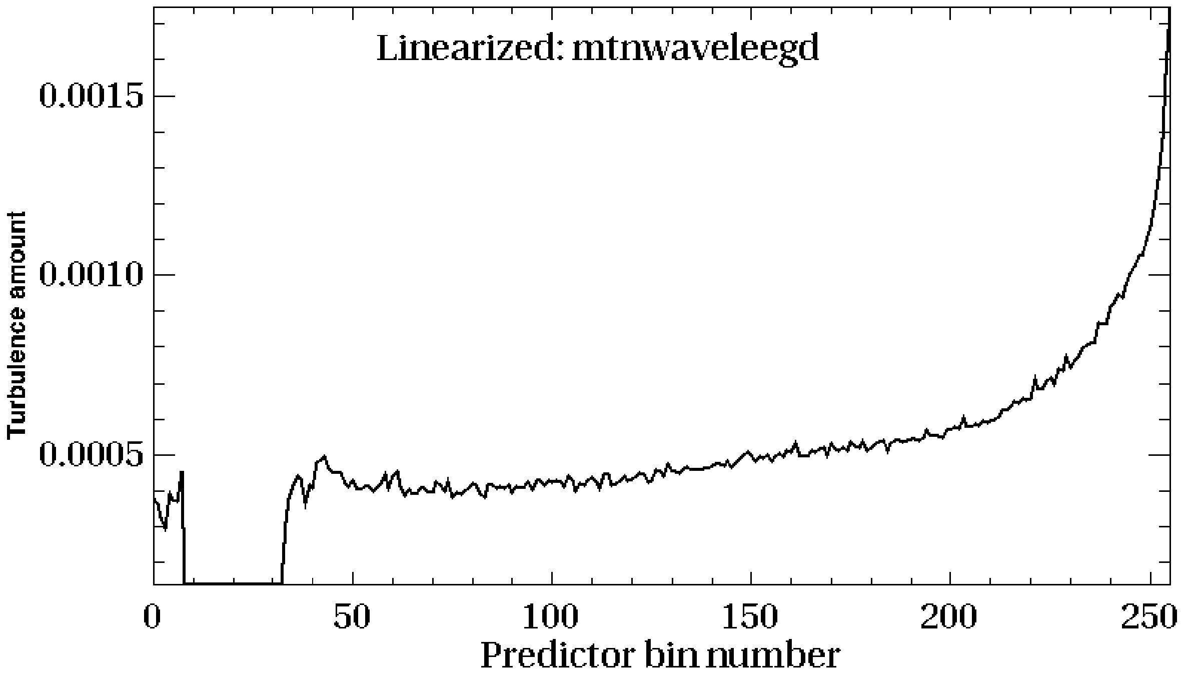


Figure B9. Mountain wave parameter, Lee filter smoother; turbulence response.

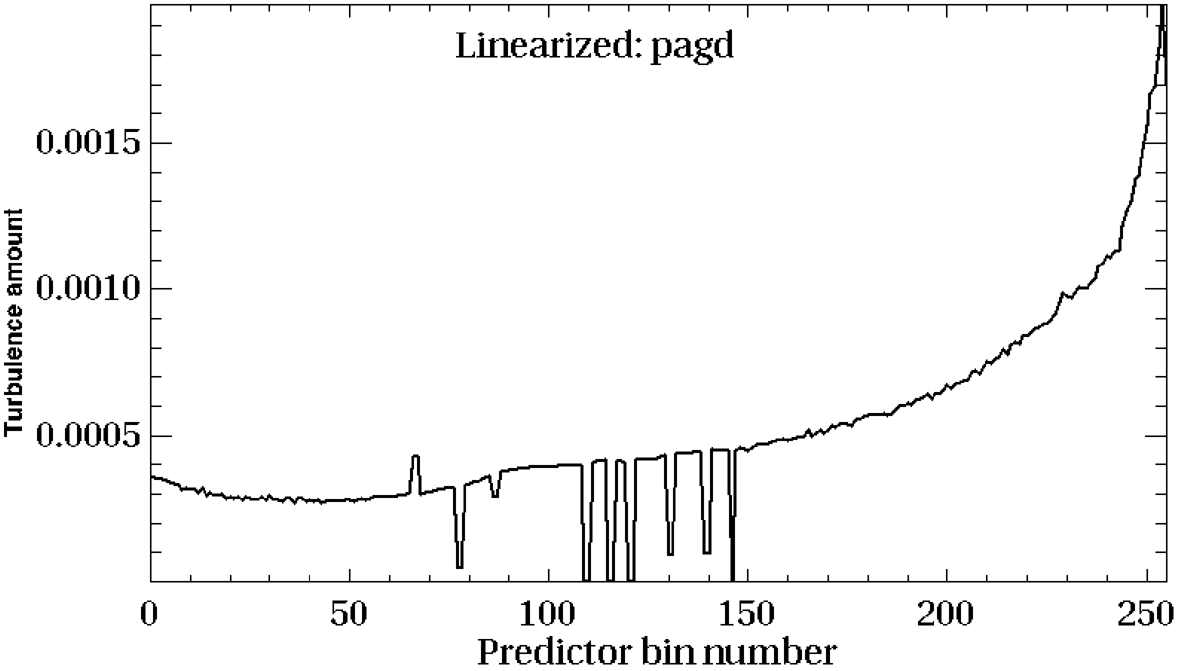


Figure B10. Panofsky index; turbulence response. Gaps in the middle are due rounding of model post-processed values of the Panofsky index, and very frequent occurrence of Panofsky index in the middle range. Thus, frequently occurring values, occurring more than 1/256 of the time, occupy the same bin, leaving adjacent bins empty. This is seen in other predictors, for example, predictors having frequent occurrences of "zero".

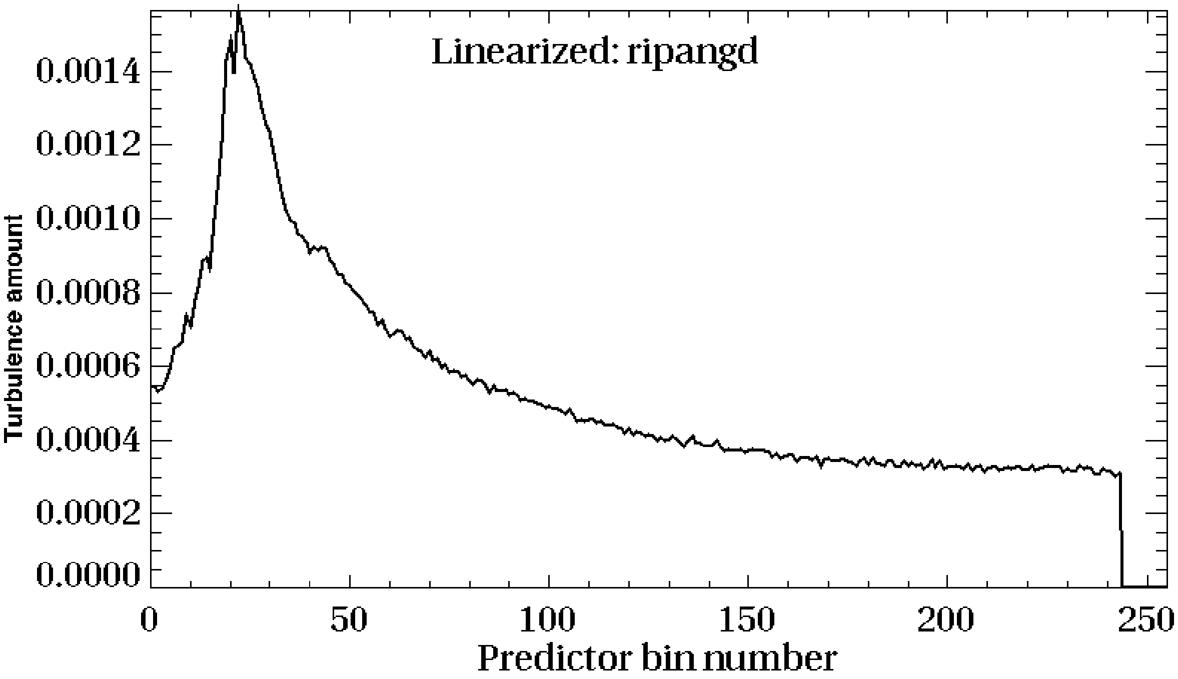


Figure B11. Richardson number variation 1; turbulence response.

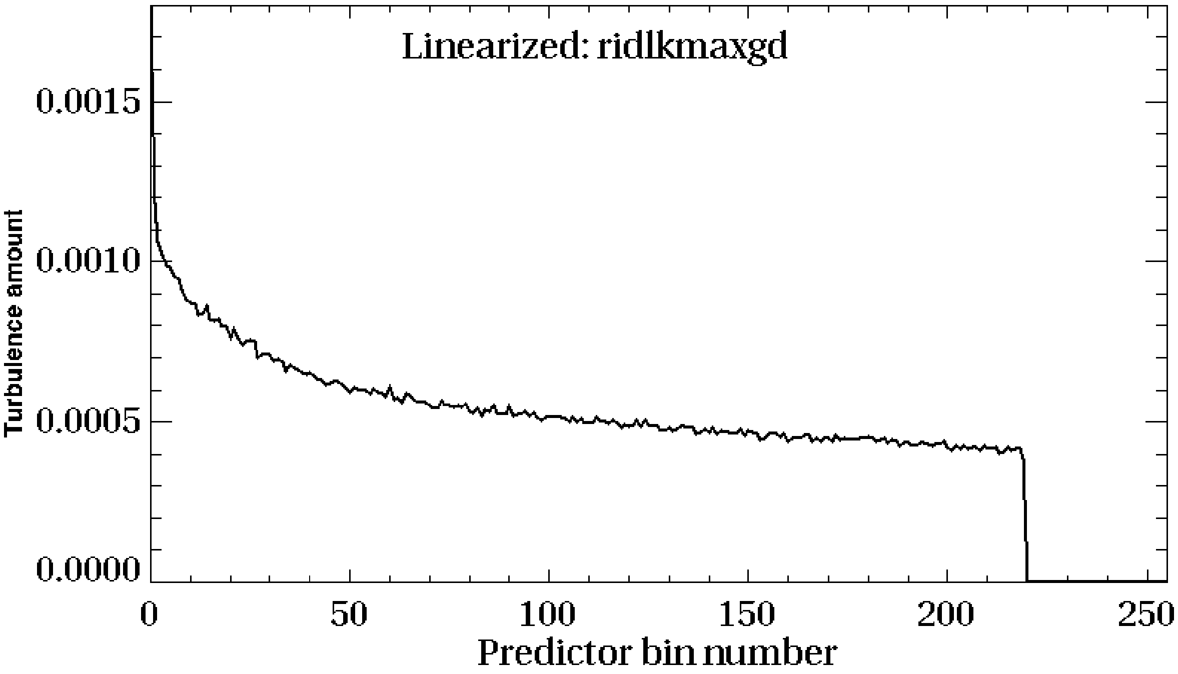


Figure B12. Richardson number variation 2; turbulence response.

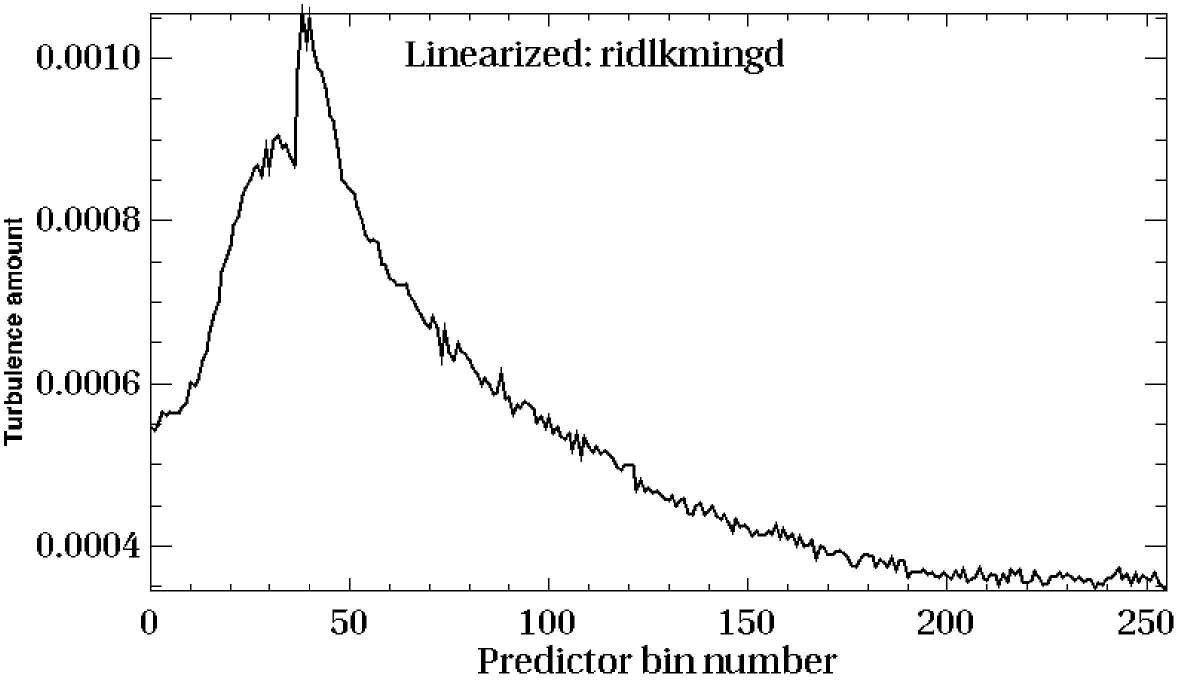


Figure B13. Richardson number variation 3; turbulence response.

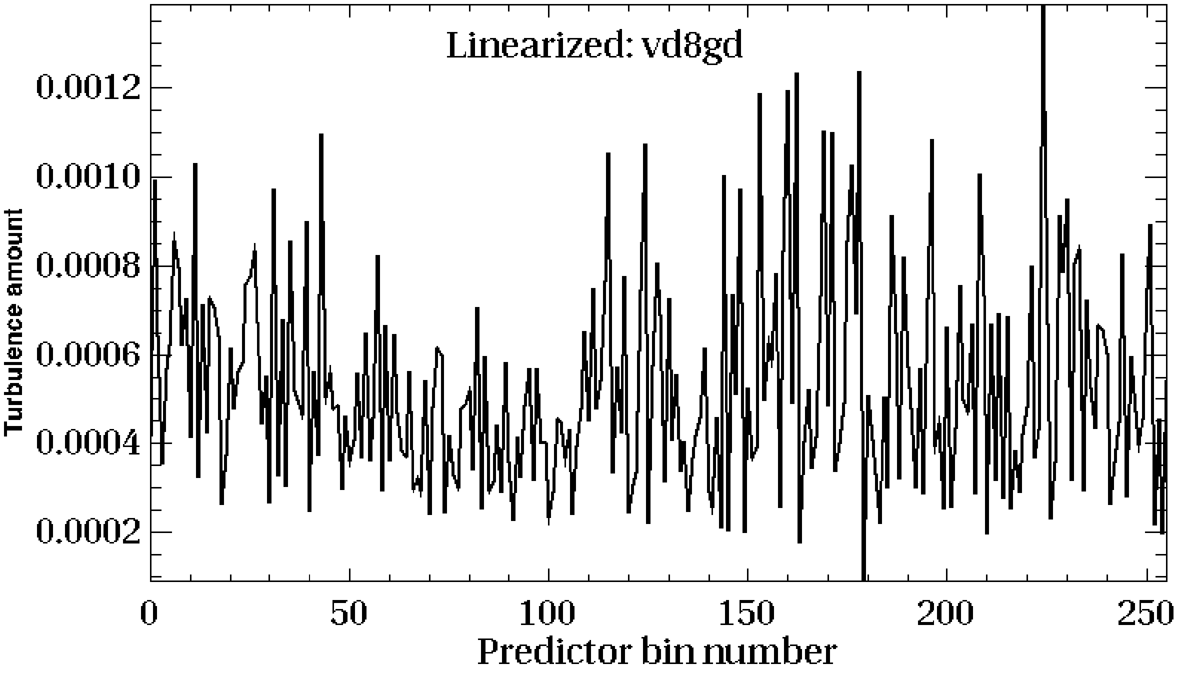


Figure B14. Year-month-day-hour of verification time ("normalized" on 0-255 scale); turbulence response.

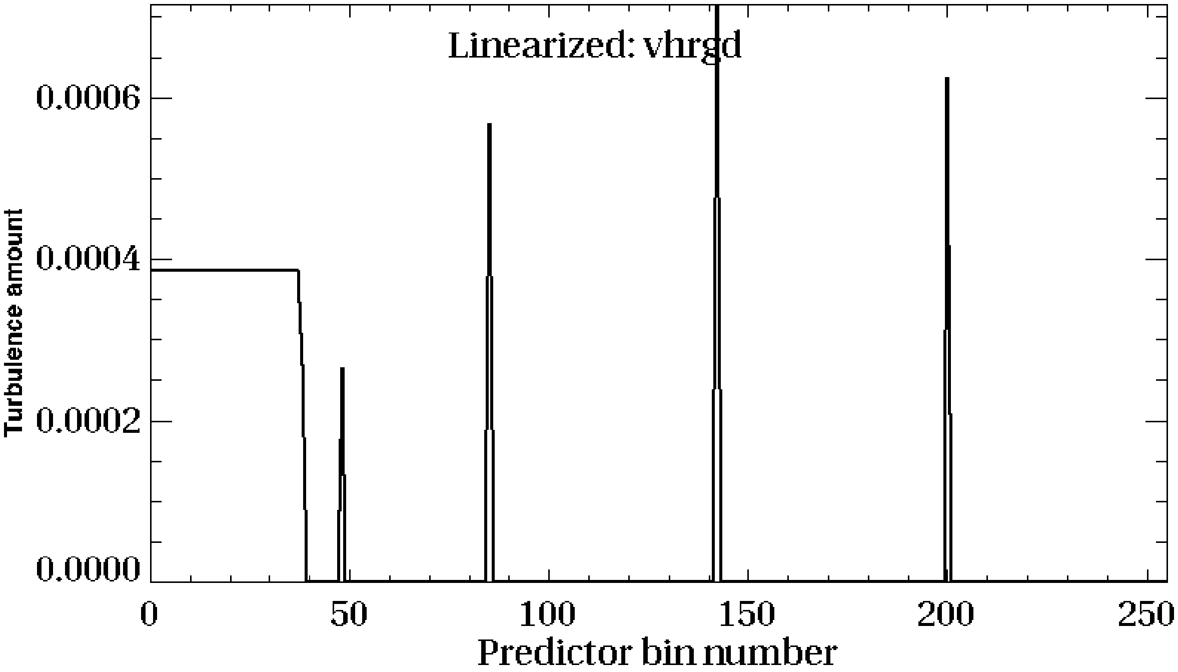


Figure B15. Hour of verification ("normalized" on 0-255 scale); turbulence response.

Table B1. Linearization of predictors to turbulence: Correlation coefficient.

Predictor	Dataset	Raw	Linearized
dyngd	Dev't	0.115	0.140
dyngd	Test	0.116	0.145
elgd	Dev't	-0.006	0.065
elgd	Test	-0.011	0.058
lapsegd	Dev't	0.054	0.061
lapsegd	Test	0.044	0.046
maxwspgd	Dev't	0.043	0.061
maxwspgd	Test	0.045	0.062
maxwspleegd	Dev't	0.060	0.068
maxwspleegd	Test	0.064	0.072
maxwspnocap	Dev't	0.061	0.079
maxwspnocap	Test	0.066	0.086
maxwspleenocap	Dev't	0.074	0.089
maxwspleenocap	Test	0.079	0.096
mtnwavegd	Dev't	0.012	0.051
mtnwavegd	Test	0.020	0.053
mtnwaveleegd	Dev't	0.062	0.089
mtnwaveleegd	Test	0.065	0.096
pagd	Dev't	0.128	0.151
pagd	Test	0.126	0.154
ripangd	Dev't	-0.112	0.137
ripangd	Test	-0.108	0.134
ridlkmaxgd	Dev't	-0.076	0.089
ridlkmaxgd	Test	-0.075	0.083
ridlkmingd	Dev't	-0.080	0.091
ridlkmingd	Test	-0.071	0.081
vd8gd	Dev't	0.019	0.121
vd8gd	Test	0.029	0.041
vhrgd	Dev't	0.039	0.052
vhrgd	Test	0.042	0.056

Appendix C

RNN technical notes for implementers

a) Computational efficiency issues

Current RF methodology is to sample approximately 2/3 of the dataset, and to sample the dataset approximately 500 times. RNN instead samples the historical dataset essentially one time (the caveat is described below). The RNN approach would appear to have a significant time advantage versus RF. Recall that RF creates ~500 randomly permuted decision trees; the strategy is to create 500 forecasts, and to use a consensus value as the final forecast. It is unclear from literature what kind of consensus is appropriate, or whether a consensus forecast is appropriate for probability forecasts. It is also unknown which, RNN or RF, create better quality forecasts.

The above paragraph states that RNN samples the archive dataset "one time". It turns out that, since RNN neighborhoods overlap, that a typical gridpoint may be sampled several times (5 times was a typical number found in one RNN run), still much less than the 500 times suggested by RF literature. The RNN strategy to select neighborhoods is as follows. All neighborhoods, dozens, hundreds, or thousands of them) area sorted in order of their Student's T-values. This sorted list of neighborhoods is used, from most significant to least significant, for a "current" forecast. Gridpoints in the target area are found that match the neighborhood. Once a forecast value is assigned to a gridpoint, that gridpoint retains its forecast value, and will not be reset by a less statistically significant neighborhood.

As described before, RNN randomly chooses gridpoints that have not yet been assigned to neighborhoods. It turns out that the last unsampled gridpoints, approximately the last 10%, become increasingly difficult to assign to neighborhoods, as they tend to consist of unusual combinations of predictor values. A casual examination of these gridpoints suggested that they are forecasts of low amounts of turbulence, specifically, less than climatology, with widely varying values of predictors. This makes sense, as the majority predictor values are unfavorable to turbulence. There are, therefore, generally more combinations of predictors that are unfavorable to turbulence than are favorable to turbulence. There is probably an opportunity to reduce the amount of categorization of turbulence forecasts by RNN where the forecast of turbulence is very low.

b) Selection of predictor bin size

A variable in the RNN process is to select an appropriate bin size. Recall that all predictors have been transformed into values ranging from 0 to 255. The same bin size was used for each predictor. This makes the implicit assumption that all predictors are equally sensitive to the predictand at all ranges of values, which is potentially untrue. An effective strategy is to begin with a relatively large bin size, such as 50, which runs more quickly. A bin size of 50 divides predictors into roughly 5 categories, which is fairly coarse. Subsequent trials will use smaller bin sizes in an effort to gain forecast skill from critical small ranges of predictor values. To measure the goodness of fit, the correlation coefficient was calculated for both the development and the test dataset. A peak value of the correlation coefficient will be found, that balances the coarse sampling of higher bin sizes, and over-sampling that will result from smaller bin sizes.

It is acknowledged that there appears to be a small amount of "overfitting" between the RNN fit to the developmental data versus the test data. As bin sizes become smaller and the number of RNN neighborhoods grows larger, the correlation coefficient of the developmental sample grows faster than the correlation coefficient of the test data. This is expected, as the development data is being over-fit. Therefore, the bin size that produces the best correlation coefficient on the test data is used as bin size that is likely to be best in the "real world". The amount of overfitting is small. The nature of the overfitting is not algebraic in nature, but the statistical risk that with more neighborhoods with fewer gridpoints, that some of the neighborhood will "just happen" to have values that are not "right". It is felt that the utilization of the so-called "test" dataset to aid in the selection of bin sizes is an adequate means of avoiding overfitting. The strategy used by the RF process is similar in that over-fit forecasts are found with the test data, however, RF utilizes a large amount of semi-random forecasts and over-sampling to detect overfitting. (TBD: this might be wrong).

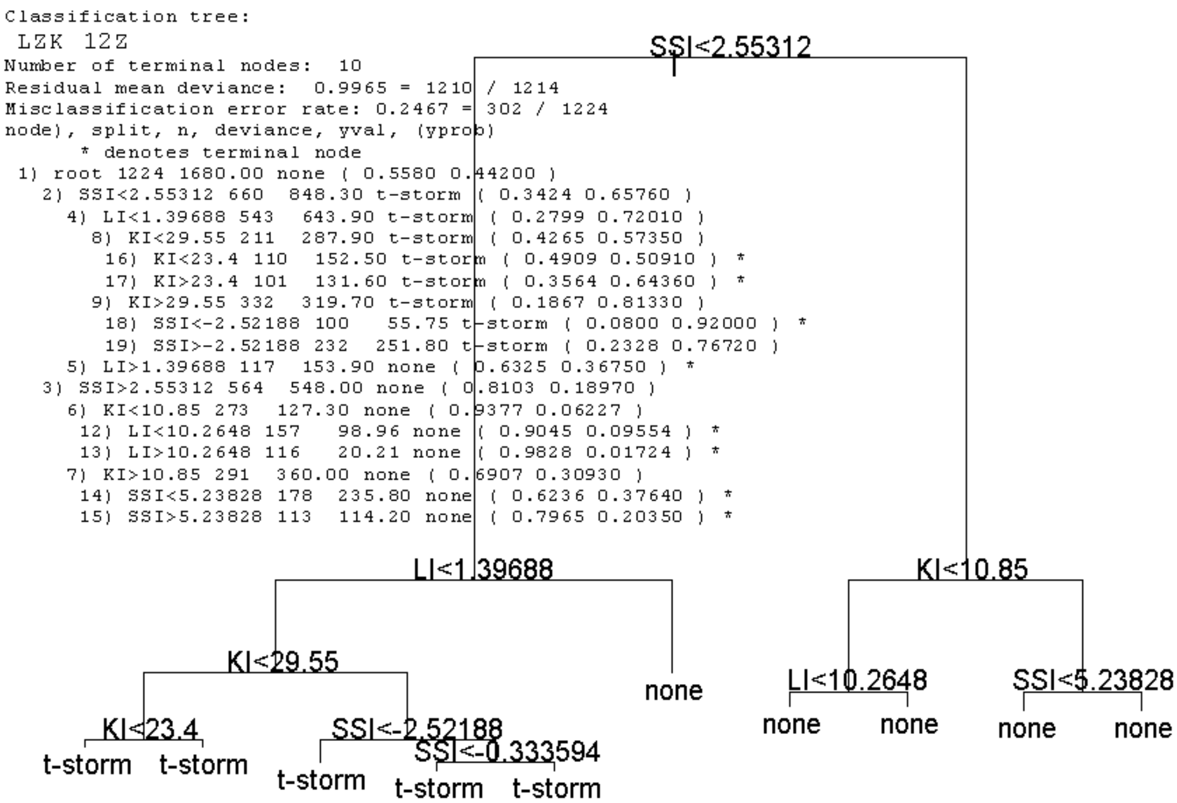
Note that with very low or very high values of a predictor (that is, predictor values near 0 and 255), the bin size may be truncated. If a gridpoint has a value of 255, and the bin size is 21, predictor values from 243 to 255 are used. The number of gridpoints per neighborhood is lessened in such cases, making them less statistically significant. However, the extreme values of predictors are often the most critical ones in forecasting weather events. "Exception coding" can be done to identify RNN neighborhoods with predictor values at the extremely favorable end of the predictor, but the Student's T-value suffers because the number of gridpoints in the neighborhood is smaller.

c) Backup forecast

A goal of the RNN process is to adequately sample the development dataset. Doing so helps to insure that a real-time forecast will have a historical precedence, that is, that a new combination of predictors will not occur in operational use. In cases where a matching neighborhood for a real time situation is not found, a backup forecast is needed: climatology, interpolation from geographically close gridpoints, or a simpler forecast. Candidates for a "simpler forecast" are the best single predictor, which is guaranteed to have a predicted value, or a two-predictor forecast, which would have a much reduced risk of a new combination of predictors being found. A more satisfying solution would be to look back into the developmental sample to create a new “neighborhood”. The impact of these previously unencountered neighborhoods on forecast scores has not been investigated.

Appendix D
Issues with decision trees

Decision trees, by their very nature, split the dataset into two portions: favorable and unfavorable to the predictand. There are multiple problems with this. Values on either side of the split, while very near to each other in value, are put into decision tree branches that have the opposite forecast (yes vs. no turbulence, for example). The decision tree then struggles to correct for values in the vicinity of the initial split. In the example from Venzke, Figure D1, making forecasts of lightning from rawinsonde data, the initial tree split is with the Showalter Index (SSI) value of +2.55. Lower in the tree, Showalter Index values of -2.52, -0.33, and +5.23 are threshold values that further refine the initial tree-split value of +2.55. Also, the Lifted Index, which is similar to the Showalter Index, is used with a splitting threshold value of +1.40.



```
Classification tree:
 LZK 12Z
Number of terminal nodes:  10
Residual mean deviance:  0.9965 = 1210 / 1214
Misclassification error rate: 0.2467 = 302 / 1224
node), split, n, deviance, yval, (yprob)
      * denotes terminal node
 1) root 1224 1680.00 none ( 0.5580 0.44200 )
   2) SSI<2.55312 660  848.30 t-storm ( 0.3424 0.65760 )
     4) LI<1.39688 543  643.90 t-storm ( 0.2799 0.72010 )
       8) KI<29.55 211  287.90 t-storm ( 0.4265 0.57350 )
        16) KI<23.4 110  152.50 t-storm ( 0.4909 0.50910 ) *
        17) KI>23.4 101  131.60 t-storm ( 0.3564 0.64360 ) *
       9) KI>29.55 332  319.70 t-storm ( 0.1867 0.81330 )
        18) SSI<-2.52188 100   55.75 t-storm ( 0.0800 0.92000 ) *
        19) SSI>-2.52188 232  251.80 t-storm ( 0.2328 0.76720 )
     5) LI>1.39688 117  153.90 none ( 0.6325 0.36750 ) *
   3) SSI>2.55312 564  548.00 none ( 0.8103 0.18970 )
     6) KI<10.85 273  127.30 none ( 0.9377 0.06227 )
      12) LI<10.2648 157   98.96 none ( 0.9045 0.09554 ) *
      13) LI>10.2648 116   20.21 none ( 0.9828 0.01724 ) *
     7) KI>10.85 291  360.00 none ( 0.6907 0.30930 )
      14) SSI<5.23828 178  235.80 none ( 0.6236 0.37640 ) *
      15) SSI>5.23828 113  114.20 none ( 0.7965 0.20350 ) *
```

Figure D1. Example decision tree output from Venzke 2001, Appendix C, page 153. Prediction of cloud to ground lightning from indices derived from the Little Rock rawinsonde.

It is clear that values of the Showalter Index and Lifted Index in the range of approximately -3 to +5 are the most sensitive to probabilities of lightning. Lower levels of the decision tree must correct for hard-set thresholds at higher levels of the tree. This example shows the need for a number of data bins in the sensitive range of a predictor, in this case, values of Showalter Index in the -3 to +5 range, with bin sizes of 1 degree Celsius or smaller. Smaller bin sizes in this critical range of values should be used to look up the probability of lightning. This is essentially the RNN approach. The decision tree attempts to do this, to divide the range of -3 to +5 into several categories, but is forced to do so using a binary tree structure.

With predictor data being split at sensitive values, the dataset might be slightly diluted, as some of the data in the sensitive range is going into the "no event" branch, and other data in the sensitive range is going into the "yes event" branch. In the Venzke example, data points with a Showalter Index value of +2 share the same initial tree branch with data points having a Showalter Index value of -5, and Showalter Index values of +3 share a tree branch with Showalter Index values of +15. In the meantime, data points with Showalter Index values of +2 and +3 are placed into different categories. This would appear to have the potential to dilute the data by categorizing similar Showalter Index values in different tree branches, and widely different values of Showalter Index into the same tree branch.

All decision tree methodologies can result in overfitting. This means that the data is divided and subdivided repeatedly, until the final tree nodes have too few data points to be statistically significant. Standard methodologies exist to mitigate this characteristic common to all decision tree techniques.

While RF attempts to address the issues listed above, it is forced to use the tree structure to do so. The RNN approach, creating neighborhoods (the equivalent of decision tree nodes) more directly, seems to be a purer approach.

The RF method of forming a forecast is to utilized some form of "consensus" forecast from 500 decision trees, either the most common value (if binary), or an average or median. This is similar to the question encountered by any ensemble methodology, which value is "the single value to use". The forecast extracted from the ensemble of forecasts naturally depends on the weather parameter that is being forecast. An unproven concern is that using a consensus from a number of decision trees, on the order of 500, may degrade the reliability of the forecast, which may trend towards climatology, and away from potentially extreme values. It is believed that RNN's neighborhood approach is a more direct approach, allowing for the more reliable forecasting of values away from the average.

Appendix E

Correlation between predictors used in this study

Table E1. Correlation between predictors used in the study.

Correlation	Pred #1 name	Pred #2 name
1	dyngd	Dyngd
-0.156	dyngd	Lapsegd
0.594	dyngd	maxwspleenocap
0.305	dyngd	mtnwaveleegd
-0.156	lapsegd	dyngd
1	lapsegd	lapsegd
-0.475	lapsegd	maxwspleenocap
0.249	lapsegd	mtnwaveleegd
0.594	maxwspleenocap	dyngd
-0.475	maxwspleenocap	lapsegd
1	maxwspleenocap	maxwspleenocap
-0.057	maxwspleenocap	mtnwaveleegd
0.305	mtnwaveleegd	dyngd
0.249	mtnwaveleegd	lapsegd
-0.057	mtnwaveleegd	maxwspleenocap
1	mtnwaveleegd	mtnwaveleegd

Appendix F
Possible reasons for overfitting with too many predictors in RNN

Recall that, using real turbulence data, RNN with four predictors showed slightly less skill than utilizing three predictors. There are four possible reasons for this. 1) The fourth predictor, the lee mountain parameter, may not be useful. The response to the lee mountain predictor, shown in Figure A-9, shows a good range of turbulence amount, from approximately .004 to .015, as good as any other predictor. There is however a rather flat response to turbulence over much of the predictor range, and this may be the reason that the RNN response is ultimately diluted with this predictor included. (2) The fourth predictor may be highly correlated with other predictors. However, the table in Appendix C shows the lee mountain parameter to have the lowest correlations to other predictors. 3) With four predictors, a large bin size was required to match four predictors at one time. This possibly degraded the fit to the data, as a bin size that is too large will not resolve turbulence responses sensitive to small predictor differences. 4) The ability of the dataset, as described, is simply inadequate to support the fitting of four predictors.